Pattern Recognition: Progress, Directions and Applications.

Edited by

Filiberto Pla
Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I, Castelló

Petia Radeva
Centre de Visió per Computador & Dept. Ciències de la Computació
Universitat Autònoma de Barcelona, Bellaterra, Barcelona

and

Jordi Vitrià
Centre de Visió per Computador & Dept. Ciències de la Computació
Universitat Autònoma de Barcelona, Bellaterra, Barcelona

Computer Vision Center,
Universitat Autònoma de Barcelona

ISBN 84-933652-6-2

Printed by Ediciones Gráficas Rey, S.L.

“Can you think of some chore or duty that a person does that she does not do better the second time? Or can you think of some chore or duty that a computer does that it does better the second time?”

Oliver Selfridge, MIT

Preface

Many Artificial Intelligence programs do not show enough intelligence so that normal people would consider them intelligent. Most of them could simply be called "advanced programming techniques", as usually they are highly useful but not intelligent programs. If we want to have a system with approximately the same ability to deal with the real world that we do then the system will need to approximate the same senses that people have and acquire most of its knowledge about the world by growing up as people do. In this context, machine learning and pattern recognition play crucial role in designing intelligent systems given that their main goal is concerned with the development of techniques which allow computers to "learn".

Pattern Recognition is a scientific field of longstanding tradition, with origins in the early years of computer science. Today, Pattern Recognition has reached a level of maturity that allows us to build highly sophisticated systems which perform very different tasks. Nevertheless, its evolution has opened up a number of new problems, ranging from specific algorithms to system integration, which remain elusive and assure a long life for this research field. The field is progressing rapidly, and an air of excitement among researchers is being created by the increasing scope of applications to which machine learning is relevant, and by the many technical advances that have been made in recent years. One reason pattern recognition is such a rapidly developing field lies in the fact that modern societies have entered the “data era”—an unprecedented investment is being made in the collection of data, with archives being formed on an enormous scale. Biological data are being collected using increasingly fast machines to scan genomes, hyperspectral satellite imagery is being stored on a massive scale, web documents are appearing at an explosive rate in internet, and so on. The development of effective ways for extracting useful information from these data stores is an overall challenge to computer science as a discipline. This goal drives much of pattern recognition and machine learning research. Pattern recognition pushed forward to development of a wide spectrum of applications like search engines, medical diagnosis, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing and robot locomotion.

In this book, we claim to give an overview of recent advances in the pattern recognition field achieved by Spanish Network on Pattern Recognition and its Applications (TIC2002-12744-E). This is a thematic network devoted to exchange and disseminate state-of-the-art research in Pattern Recognition among research groups within Spain and across the rest of Europe. Member groups are working on challenging theoretical projects (multiple classifiers, feature selection, prototype classification, distance-based approaches, error-correcting output codes, hierarchical clustering, Bayes models, etc.) as well as

advanced applications of the pattern recognition field in different artificial intelligence projects: speech recognition, biometric verification, hyperspectral image classification, genre recognition, video-based face processing, medical endoscopy motility indexing, etc. From its beginning the network created an excellent environment for scientific brainstorming, organized several scientific meetings with world-wide well-known scientists in the field of Pattern Recognition, created common databases and challenges, and developed an excellent and exciting environment for research and development. The following book containing 23 chapters gives an overview of the scientific activity of all groups that actively participated and contributed to the life of the network.

We would like to thank all the authors for their help in the editing process. It is their competence and work which has enabled the editors to put together this book.

Filiberto Pla, Petia Radeva and Jordi Vitrià

Bellaterra & Castelló, March 2006

Contents

Pattern Recognition approaches to Machine Translation and Computer Assisted Translation at PRHLT group *

J. Andrés F. Casacuberta J. Civera E. Cubel
I. García-Varea J. González M. T. González A. L. Lagarda
J. R. Navarro F. Nevado D. Ortiz D. Picó L. Rodríguez
G. Sanchis J. Tomás E. Vidal
J. M. Vilar
Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia,
Camino de Vera, s/n - 46022 Valencia, Spain.
prhlt@iti.upv.es

Abstract

Machine Translation technologies are becoming increasingly important in a globalized world. PRHLT group has successfully applied pattern recognition techniques in several projects involving both text and speech translation.

A new and very promising approach is to use a computer to assist a human translator, thereby joining the power of computers with human expertise. In this regard, PRHLT group has concluded an important Computer Assisted Translation project, creating a translation engine capable of incorporating the corrections made by human during the translation process. This ensures high quality translations and productivity improvements.

In addition, PRHLT group was the leader of another Speech Translation European project.

Keywords: Machine Translation, Computer-Assisted Translation.

1 Introduction

PRHLT is a research group specialized in the research field of Pattern Recognition and Natural Language Processing, hence its name: Pattern Recognition and Human

* Work supported by the *A*gencia Valenciana de Ciencia y Tecnología (AVCiT) under grant GRUPOS03/031, the European Union under the IST Programme (IST-2001-32091), the Spanish CICYT under grant TIC2003-08681-C02-02 and the AMETRA project (INTEK-CN03AD02).

Language Technology. The group is formed by eleven Ph.D.'s and eighteen Ph.D. students, from which fifteen are professors and assistants and the rest are research contracts and fellowships.

The main research lines addressed at PRHLT group are: Language Translation, Speech Recognition, Handwritten Character Recognition, Biometrics and Computer Vision. More specifically, the group has been specially active during the last few years in the field of Machine Translation, accomplishing to report important contributions to the scientific community.

Here, we present the most important achievements of the group, divided into four sections. In sections 2 and 3 we give a short introduction to the state-of-the-art Machine Translation and Computer Assisted Translation, describing the main research lines within these fields.

In section 4, we present the main projects in which our group has taken part, describing their main purpose and their most important results.

Finally, in section 5 we list a short summary of the main papers published by the group.

2 Machine Translation

2.1 Introduction

Machine translation (MT) is an important area to the European Union and the Information Society Technologies. A breakthrough in this area would have an important socio-economic impact. The development of a classical MT system requires a great human effort.

Two main approaches to MT exist, based on linguistic or statistical methods. Machine translation can be tackled from a linguistic point of view. In this way, two big families exist: *knowledge-based* and *corpus-based* methods. Knowledge-based techniques formalize expert linguistic knowledge, in form of rules, dictionaries, etc., in a computable way. Corpus-based methods use statistical pattern recognition techniques to automatically infer models from bilingual text samples without necessarily using a-priori linguistic knowledge. In addition to linguistic methods, *Statistical machine translation* (SMT) has proved to be an interesting framework where MT systems can be built (quasi) automatically if adequate parallel corpora are available [1].

The MT problem can be statistically stated as follows: given a sentence $\mathbf{s}$ from a source language, search for a target-language sentence $\hat{\mathbf{t}}$ which maximises the

posterior probability[1]:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \Pr(\mathbf{t}|\mathbf{s}) \ . \tag{1}$$

There are some approaches to SMT, one of them is based on two statistical models: a *target (statistical) language model* and a *translation model*. It is commonly accepted that a convenient way to deal with Eq. 1 is to transform it by using the Bayes' theorem [1]:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s}|\mathbf{t}) \ , \tag{2}$$

where $\Pr(\mathbf{t})$ is aproximated by a *target language model*, which gives high probability to well formed target sentences and $\Pr(\mathbf{s}|\mathbf{t})$ accounts for source-target word(-position) relations and is based on *stochastic dictionaries* and *alignment models*.

The widely used target language model is the (smoothed) n-gram: let I be the length of a target sentence $\mathbf{t}$,[2]

$$\Pr(\mathbf{t}) \approx \prod_{i=1}^{I} p(\mathbf{t}_i \mid \mathbf{t}_{i-n+1}^{i-1}) \ , \tag{3}$$

where the probability of a target word $\mathbf{t}_i$ depends on the last $n-1$ words $\mathbf{t}_{i-n+1}^{i-1}$.

There are different proposals as translation models and the first ones were based on *single-word* (SW) alignment models [2]. In this case, the basic assumption is that each source word is generated by only one target word. This assumption does not correspond to the nature of natural language; in some cases, we need to know the context of the word to be translated. One way to upgrade this simple assumption is the use of statistical context-dependent dictionaries as in [3]. Another way to overcome the above-mentioned restriction of single-word models is known as the *template-based* (TB) approach [4]. In this approach, an entire group of adjacent words in the source sentence may be aligned with an entire group of adjacent target words.

Recent works present a simple alternative to these models, the *phrase-based* (PB) approach [5, 6]. In these models each sequence of words in the source sentence is translated into another sequence of words into the target sentence with a certain probability. *Maximum Entropy*, firstly introduced in this field by [7], and *Recursive Alignment* techniques [8] have been also applied to capture the contextual information.

[1] For simplicity, $\Pr(X = x)$ and $\Pr(X = x \mid Y = y)$ are denoted as $\Pr(x)$ and $\Pr(x \mid y)$.

[2] Following a notation used in [2], a sequence of the form $z_i, \ldots, z_j$ is denoted as z_i^j. For some positive integers N and M, the image of a function $f : \{1, ..., N\} \rightarrow \{1, ..., M\}$ for n is denoted as f_n, and all the possible values of the function as f_1^N

An alternative to Eq. 2 is to transform Eq. 1 differently:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \Pr(\mathbf{s}, \mathbf{t}) \ . \tag{4}$$

In this case, the joint probability distribution can be adequately modelled by means of Stochastic Finite State Transducers (SFST) [9]. These models can deal with some source and target syntactic restrictions together with the relation between sequences of source words and sequences of target words [10]. On the other hand, they can be applied also for *speech translation* in a similar way as n-gram is used for speech decoding [11].

Usually, MT systems take as input a text in a source language and translate it into a text in a target language. Nonetheless, the translation process cannot only be text-to-text but also speech-to-speech. This speech-to-speech translation will be explained in the last point of this section.

2.2 Statistical Alignment Models

In the following sections, state-of-the-art statistical alignment models will be described, including *IBM* translation models, *Phrase Based* translation models, *Maximum Entropy* models and *Recursive Alignment* models.

2.2.1 IBM Translation Models

In [2] the so-called *IBM* models are proposed, which are a possible way of estimating the translation model within SMT. These models are based on the concept of alignment between the components of the *translation pairs*.

Let s_1^J and t_1^I be some source and target sentences of length J and I, respectively. Formally, an alignment is a mapping between the sets of positions in s_1^J and t_1^I: $\mathbf{a} = a_1^J \subseteq \{1 \cdots J\} \times \{1 \cdots I\}$. Alignment models to structure the translation model are introduced in [2]. These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. In statistical alignment models, $Pr(s_1^J, a_1^J | t_1^I)$, the alignment a_1^J is introduced as a hidden variable.

The translation probability $Pr(s_1^J, a_1^J | t_1^I)$ can be rewritten as follows:

$$\begin{aligned} Pr(s_1^J, a_1^J | t_1^I) &= \prod_{j=1}^{J} Pr(s_j, a_j | s_1^{j-1}, a_1^{j-1}, t_1^I) \\ &= \prod_{j=1}^{J} \Big(Pr(a_j | s_1^{j-1}, a_1^{j-1}, t_1^I) \cdot Pr(s_j | s_1^{j-1}, a_1^j, t_1^I) \Big) \ . \end{aligned} \tag{5}$$

Five *IBM* models exist, from *IBM* model 1 to *IBM* model 5, of increasing complexity, which model in a different way the relation between source and target words.

Learning *IBM* statistical alignment models

All the necessary for estimating *IBM* alignment models is described in [2]. The estimation of the parameters of these models is carried out by maximum-likelihood estimation via the EM algorithm [12]. The public available tool GIZA++ [13] is a possible implementation to perform this estimation.

Search

Given a source sentence s_1^J, the aim of the search in statistical machine translation is to look for a target sentence $\hat{\mathbf{t}}$ that maximises the product $\Pr(t_1^I) \cdot \Pr(s_1^J|t_1^I)$. Different algorithms have been proposed to searching with *IBM* models. The basic idea of most of these algorithms is to generate partial hypotheses about the target sentence in an incremental way. Each of these hypotheses is composed by a prefix of the target sentence, a subset of source positions that are aligned with the positions of the prefix of the target sentence and a score. New hypotheses can be generated from a previous hypothesis by adding a target word(s) to the prefix of the target sentence that is (are) the translation of a source word(s) that is (are) not translated yet.

The search process, when using a statistical alignment model and depending on the search algorithm used, yields to search criteria. In general, eq. 2 can be rewritten as follows:

$$\begin{aligned} \hat{\mathbf{t}} &= \arg\max_{\mathbf{t}} \{Pr(\mathbf{t}) \cdot Pr(\mathbf{s}|\mathbf{t})\} \\ &= \arg\max_{\mathbf{t}} \left\{ Pr(\mathbf{t}) \cdot \sum_{\mathbf{a}} Pr(\mathbf{s}, \mathbf{a}|\mathbf{t}) \right\} \end{aligned} \tag{6}$$

and using the maximum aproximation we have:

$$\begin{aligned} \hat{\mathbf{t}} &\approx \arg\max_{\mathbf{t}} \left\{ Pr(\mathbf{t}) \cdot \max_{\mathbf{a}} Pr(\mathbf{s}, \mathbf{a}|\mathbf{t}) \right\} \\ &= \arg\max_{\langle \mathbf{t}, \mathbf{a} \rangle} \{Pr(\mathbf{t}) \cdot Pr(\mathbf{s}, \mathbf{a}|\mathbf{t})\} \end{aligned} \tag{7}$$

Different search strategies have been proposed to define the way in which the search space is organized. Namely *Stack-Based Decoding* algorithms, *Dynamic Programming* algorithms and *Greedy Decoding* algorithms.

- **Stack-Based Decoding**

  Stack Decoding algorithm, also called $A*$ algorithm [14], attempts to generate partial solutions or hypotheses, until a complete sentence is found. These hypotheses are stored in a stack and sorted using a *score*[3]. Typically, this measure is a probability value given by both the translation and the language models. The decoder follows a *best-first* strategy in order to achieve an optimal hypothesis:

  1. Initialization of the stack with an empty hypothesis
  2. Iteration
     (a) Pop h (the best hypothesis) off the stack
     (b) If h is a complete sentence, output h and end
     (c) Expand h
     (d) Go to step *2a*

  A *depth-first* strategy can also be employed, as in [15] that utilises a set of stacks in order to perform the search. Concretely, the algorithm uses a different stack to store the hypothesis depending on which words in the source sentence have been translated. This procedure allows to force the expansion of hypotheses with a different degree of completion. In each iteration, the algorithm covers all stacks with some hypotheses and extends the best one for each.

- **Dynamic Programming**

  Dynamic programming [16, 17] creates a table of solutions to all subproblems that might occur. It is based on the principle of optimality, where the traditional term *policy* is used for a decision rule that determines the next state given a predecessor state. This optimality principle is stated as follows: an optimal policy has the property that, whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

- **Greedy Decoding**

  Greedy decoders were proposed in [14] for the first time. The main difference between this search algorithm and the ones presented in previous sections is that it does not follow an incremental process to build an output hypothesis. It starts from an initial complete hypothesis and iterates a process in which,

[3] The score is actually the log of the resulting probability.

in every iteration, an operation (or transformation) is applied to the current hypothesis to obtain a better one. The initial hypothesis is constructed by choosing the best inverse translation of every word of the input sentence. This initialization method provides a monotone alignment. A different initialization method which obtains better results is proposed in [17] where the Viterbi alignment (for a specific model) is also applied.

2.2.2 Phrase-based alignment models

In SW alignment models, words are translated individually, without considering the context. PB alignment models constitute an interesting and simple alternative that allows to model this contextual information [5, 6]. The principal innovation of these methods is that they attempt to calculate the translation probabilities of word sequences (phrases) rather than only single words.

One shortcoming of the PB alignment models is its generalization capability, since only sequences of segments that have been seen in the training corpus are accepted.

The derivation of the PB models is based on the concept of bilingual segmentation, i.e. sequences of source words and target words. It is assumed that only segments of contiguous words are considered, the number of source segments is the same as the number of target segments (say K) and each source segment is aligned with only one target segment and vice versa.

The main estimation technique of these models is based on single-word alignments, usually obtained from the public available software GIZA++ [18]. *Thot Toolkit* [19], developed at PRHLT, performs this kind of estimation.

The search process described in 2.2.1 for *IBM* models is very similar to the one for PB models. Here we can also adopt a *depth-first* strategy as defined in [20]. Additionally, a dynamic programming based search is described in [21], which is implemented in the public available tool called *Pharaoh* [22].

2.2.3 Maximum Entropy models

Current statistical machine translation systems are mainly based on statistical word lexicons. However, these models are usually context-independent, therefore, the disambiguation of the translation of a source word must be carried out using other probabilistic distributions (distortion distributions and statistical language models). One efficient way to add contextual information to the statistical lexicons is based on maximum entropy modeling [23]. In that framework, the context is introduced through feature functions that allow us to automatically learn context-dependent

lexicon models.

In a first approach [7], maximum entropy modeling is carried out after a process of learning standard statistical models (alignment and lexicon). In a second approach [24], the maximum entropy modeling is integrated in the expectation-maximization process of learning standard statistical models.

2.2.4 MAR

MAR [8] is designed so that the alignment between two sentences can be seen in a structured manner: each sentence is divided into two parts and they are put in correspondence; then each of those parts is similarly divided and related to its translation. In this way, the alignment can be seen as a tree structure which aligns progressively smaller segments of the sentences. This recursive procedure gives its name to the model: MAR, which comes from "Modelo de Alineamiento Recursivo", which is Spanish for "Recursive Alignment Model".

IBM model 1 is not adequate to describe complex translations in which complicated patterns and word order changes may appear. Nevertheless, this model can do a good job to describe the translation of short segments of texts.

To overcome that limitation of the model the following approach will be taken: if the sentence is complex enough, it will be divided in two and the two halves will be translated independently and joined later; if the sentence is simple, the *IBM* model 1 will be used.

2.3 Finite State Transducers Inference

Stochastic Finite-State Transducers (*SFSTs*) can be learned automatically from bilingual sample pairs. SFSTs are finite-state networks that accept sentences from a given input language and produce sentences of an output language.

A particular case of finite-state transducers are known as subsequential transducers (SSTs) [25]. These are essentially finite-state transducers with the restriction of being deterministic. The main advantage of SST relies on allowing typical word reorderings in the translated sentence in order to guarantee a correct output. This is possible because of SST are able to delay the emission of output symbols until the corresponding prefix of the input sentence is parsed/analyzed.

OSTIA and OMEGA algorithms are devoted to the automatic generation of SSTs. The first of this two algorithms is based only in finite-state techniques while OMEGA is a hybrid method that combines this techniques with some additional information extracted from statistical methods. Finally, a third hybrid method called GIATI that infers SFSTs will be studied.

2.3.1 An Onward Subsequential Transducer Inference Algorithm: OSTIA

Given a set of training sentence pairs, the OSTIA efficiently learns a SST that generalises the training set [25]. The algorithm builds a straightforward prefix-tree representation of all the training pairs and moves the output strings toward the root of this tree as much as possible, leading to an "onward" tree representation. Finally a state merging process is carried out. The algorithm guarantees identification of total subsequential functions in the limit, that is, if the unknown target translation exhibits a subsequential structure, convergence to it is guaranteed whenever the set of training samples is representative.

Nevertheless, there are *partial* subsequential functions for which OSTIA inference is troublesome. This limitation can be solved by an extension, called OSTIA-DR (OSTIA with Domain and Range constraints) [26] in which the learnt transducers only accept input sentences and only produce output sentences compatible with the input/output language models.

2.3.2 Hybrid (statistical/finite-State) inference algorithms

An inconvenience of finite-state transducer learning techniques like OSTIA (and all its extensions) is that they seem to require large amounts of training data to produce adequate results. However, some byproducts of statistical translation models [1] can be useful to improve the learning capabilities of finite-state models.

OMEGA

The OMEGA (for the Spanish OSTIA *Mejorado Empleando Garantías y Alineamientos*) [27] algorithm is an extension of the OSTIA algorithm that incorporates some additional information extracted from statistical translation models into the learning process. Specifically, it allows the use of statistical dictionaries and alignments estimated from the same training pairs that will be employed by OMEGA. These stochastic dictionaries and alignments establish input-to-output, word and word position relationships that enrich OSTIA algorithm. In the present work, these statistical translation models were estimated using the GIZA++ toolkit [13], which implements *IBM* statistical models [1].

An stochastic extension of OMEGA, called OMEGA-P, can be defined with the same transition and final state probability estimation strategy than OSTIA-P.

GIATI

An algorithm for learning SFSTs is the GIATI technique [10]. Given a finite sample of string pairs, it works in three steps:

1. Building training strings. Each training pair is transformed into a single string

from an extended alphabet to obtain a new sample of strings.

2. Inferring a (stochastic) regular grammar. Typically, a smoothed n-gram language model is inferred from the set of strings obtained in the previous step.
3. The transformation of the inferred regular grammar into a transducer is trivial. The symbols associated to the grammar rules are replaced by source/target symbols, thereby converting the grammar inferred in the previous step into a transducer.

The transformation of a parallel corpus into a single string corpus is performed using statistical alignments. As in the OMEGA algorithm, these statistical alignments were calculated with the GIZA++ toolkit.

2.4 Speech to Speech translation

From a formal point of view, the problem of *speech to speech translation* (S2ST) can be stated as follows: given an utterance $\mathbf{x}$ from a source language, we have to search for a target sentence $\hat{\mathbf{t}}$ for which the next posterior probability is maximum:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{x}) \ . \tag{8}$$

The most crude approximation consists in using a conventional speech recognition system to decode $\mathbf{x}$ into a sentence $\hat{\mathbf{s}}$ from the source language [28]:

$$\hat{\mathbf{s}} = \underset{\mathrm{s}}{\operatorname{argmax}} \Pr(\mathbf{s} \mid \mathbf{x}) = \underset{\mathrm{s}}{\operatorname{argmax}} \Pr(\mathbf{s}) \cdot \Pr(\mathbf{x} \mid \mathbf{s}) \ , \tag{9}$$

where a *n-gram* or a *stochastic finite-state automaton* is used as a source language model to estimate $\Pr(\mathbf{s})$ and *hidden Markov models* (HMM) are used as acoustic models to estimate $\Pr(\mathbf{x} \mid \mathbf{s})$ [29, 28]. Once $\hat{\mathbf{s}}$ is obtained, it is used as the given $\mathbf{s}$ in Eq. 2 or Eq. 4 to obtain $\hat{\mathbf{t}}$. This is often referred to as a *"serial"* or *"loosely coupled"* S2ST approach.

In a more formal framework [30], every possible decoding of a source utterance $\mathbf{x}$ is considered as the value of a hidden variable $\mathbf{s}$. Correspondingly, Eq. 8 can be rewritten as:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \sum_{\mathrm{s}} \Pr(\mathbf{t}, \mathbf{s} \mid \mathbf{x}) \ . \tag{10}$$

If it is further assumed that $\Pr(\mathbf{x} \mid \mathbf{s}, \mathbf{t})$ does not depend on $\mathbf{t}$ (which does not always

hold, but it is reasonable in practice [30]), from Eq. 10 we obtain:

$$\hat{\mathbf{t}} = \underset{\mathrm{t}}{\operatorname{argmax}} \sum_{\mathrm{s}} \Pr(\mathbf{s}, \mathbf{t}) \cdot \Pr(\mathbf{x} \mid \mathbf{s}) \tag{11}$$

$$= \underset{\mathrm{t}}{\operatorname{argmax}} \sum_{\mathrm{s}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{x} \mid \mathbf{s}) \ . \tag{12}$$

The sum in Eq. 11 can be approximated by a maximization, $\Pr(\mathbf{s}, \mathbf{t})$ can be modeled by a SFST and HMMs can be used for modeling $\Pr(\mathbf{x} \mid \mathbf{s})$. This way, the acoustic models can be easily embedded in the translation model, yielding a Viterbi-based quasi-optimal *"integrated"* or *"tightly coupled"* approach to S2ST [11]. This approach has been deeply explored in the EuTrans project (see section 4.1).

3 Computer-assisted translation

State-of-the-art machine translation (MT) techniques are still far from producing high quality translations. This drawback leads us to introduce an alternative approach to the translation problem that brings human expertise into the MT scenario. This idea was proposed in [31] and can be illustrated as follows. Initially, the human translator is provided with a possible translation for the sentence to be translated. Unfortunately in most of the cases, this translation is not perfect, so the translator amends it and asks for a translation of the part of the sentence still to be translated (completion). This latter interaction is repeated as many times as needed until the final translation is achieved.

The scenario described in the previous paragraph can be seen as an iterative refinement of the translations offered by the translation system, that while not having the desired quality, can help the translator to increase his/her productivity. Nowadays, this lack of translation excellence is a common characteristic in all machine translation systems. Therefore, the human-machine synergy represented by the Computer-Assisted Translation (CAT) paradigm seems to be more promising than fully-automatic translation in the near future.

The CAT approach has two important aspects: the models need to provide adequate completions and they have to do so efficiently under usability constrains. To fulfill these two requirements, *stochastic finite-state transducers* (SFST) and *phrase-based* (PB) models have proved in the past to be able to provide adequate translations. In addition, it is shown in this paper that efficient searching algorithms can be easily adapted in order to provide completions (rather than full translations) in a very efficient way.

Interactive search

The concept of interactive search is closely related to the CAT paradigm. This paradigm introduces a new factor $\mathbf{t}_p$ into the general MT equation (Eq. 1). $\mathbf{t}_p$ represents a prefix of the target sentence obtained as a result of the interaction between the human translator and the MT system.

In each iteration, a prefix ($\mathbf{t}_p$) of the target sentence has somehow been fixed by the human translator in the previous iteration and the CAT system computes its best (or n-best) translation suffix hypothesis ($\hat{\mathbf{t}}_s$) to complete this prefix.

Given $\mathbf{t}_p\hat{\mathbf{t}}_s$, the CAT cycle proceeds by letting the user establish a new, longer acceptable prefix. To this end, he or she has to accept a part ($\mathbf{a}$) of $\mathbf{t}_p\hat{\mathbf{t}}_s$ (or, more typically, just a prefix of $\hat{\mathbf{t}}_s$). After this point, the user may type some keystrokes ($\mathbf{k}$) in order to amend some remaining incorrect parts. Therefore, the new prefix (typically) encompasses $\mathbf{t}_p$ followed by the accepted part of the system suggestion, $\mathbf{a}$, plus the text, $\mathbf{k}$, entered by the user. Now this prefix, $\mathbf{t}_p\mathbf{a}\,\mathbf{k}$, becomes a new $\mathbf{t}_p$, thereby starting a new CAT prediction cycle.

Ergonomics and user preferences dictate exactly when the system can start its new cycle, but typically, it is started after each user-entered word or even after each new user keystroke.

Perhaps the simplest formalization of the process of hypothesis suggestion of a CAT system is as follows. Given a source text $\mathbf{s}$ and a user validated *prefix* of the target sentence $\mathbf{t}_p$, search for a *suffix* of the target sentence that maximises the *a posteriori* probability over all possible suffixes:

$$\hat{\mathbf{t}}_s = \underset{\mathrm{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{t}_p) \ . \tag{13}$$

Taking into account that $\Pr(\mathbf{t}_p \mid \mathbf{s})$ does not depend on $\mathbf{t}_s$, we can write:

$$\hat{\mathbf{t}}_s = \underset{\mathrm{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_p\mathbf{t}_s \mid \mathbf{s}) \ , \tag{14}$$

where $\mathbf{t}_p\mathbf{t}_s$ is the concatenation of the given prefix $\mathbf{t}_p$ and a suffix $\mathbf{t}_s$. Eq. 14 is similar to Eq. 2, but here the maximisation is carried out over a set of suffixes, rather than full sentences as in Eq. 2. This joint distribution can be adequately modeled by means of SFSTs [32].

The main critical aspect of the interactive CAT system is the response time. To deal with this issue, one solution is the use of a word graph. Therefore, the above mentioned maximisation problem has been devised in two phases. The first one copes with the extraction of a word graph $\mathcal{W}$ from a SFST $\mathcal{T}$ given a source

sentence $\mathbf{s}$. In a second phase, the search of the best translation suffix (or suffixes) is performed over the word graph $\mathcal{W}$ given a prefix $\mathbf{t}_p$ of the target sentence.

4 Research projects

In the following sections, some of the main research projects developed at PRHLT will be summarized.

4.1 EUTRANS:Example-Based Language Translation Systems

The EuTrans project (ESPRIT-LTR Project Number 30.268) is aimed at using Example-Based (EB) techniques for developing MT systems for limited-domain tasks which require text and/or speech input.

EuTrans was planned as a two stages project. In the first phase basic demonstration systems for text-input and speech-input translation were developed. These prototypes rely on example-based (EB) techniques for learning a kind of finite-state translation models, known as Subsequential Transducers. These models lend themselves particularly well to being integrated with acoustic-phonetic, lexical and syntactic models in order to perform speech-input MT. This allows the building of systems in which all the models required for each new application are automatically learned from training data.

Following the successful paradigm started in the first phase, particular attention was paid to the tight integration of translation and speech recognition, with the aim of achieving a high degree of robustness in speech-input operation.

Relevant results

The feasibility of these EB techniques and their usefulness from a final-user perspective were demonstrated by building both text-input and speech-input prototypes following user-centered methodologies.

Important benefits were obtained from the results:

- Example-Based approaches allow for automatically building MT systems from training examples of each considered task. This reduces the development costs of MT systems in many specific domains, as compared with more traditional Knowledge-Based approaches.

- Tight integration of translation and speech models allows for low-cost, real-time, robust speech-input translation for limited-domain applications, in contrast with the comparatively high computational demands often entailed by

more conventional approaches based on a significantly less robust "first-recognition then-translation" paradigm.

4.2 SISHITRA:Hybrid translation systems from Valencian to Castilian

SisHiTra (Hybrid Translation System) project (CICYT FEDER TIC2000-1599-C02) that combines knowledge-based and corpus-based techniques to produce a Spanish-to-Catalan machine translation system with no semantic constraints. Spanish and Catalan are languages belonging to the Romance language family and have a lot of characteristics in common. SisHiTra makes use of their similarities to simplify the translation process. A SisHiTra future perspective is the extension to other language pairs (Portuguese, French, Italian, etc.).

Knowledge-based techniques are classical approaches to tackle general scope machine translation systems. Nevertheless, inductive methods have shown competitive results dealing with semantically constrained tasks.

Relevant results

Innovative methodologies have been used to represent the different knowledge sources, such as disambiguation modules based on Hidden Markov Models or dictionaries taking advantage of stochastic transducers.

4.3 AMETRA:Computer assisted translation based on translation memories

The goal of the AMETRA project (INTEK-CN03AD02) was to make a computer-assisted translation tool from the Spanish language to the Basque language under the memory-based translation framework. The system is based on a large collection of bilingual word-segments. These segments are obtained using linguistic or statistical techniques from a Spanish-Basque bilingual corpus consisting of sentences extracted from the Basque Country's official government record.

Relevant results

The success of a statistical machine translation system relies on the availability of a large bilingual corpus to be used to train different translation and language models. Thus, is specially important the quality of such a corpus in terms of complexity. Ideally, the corpus should be perfectly split into sentences, be free of noise and errors and be free as possible of incorrect translations. In practice, this is not usually the case. New corpora usually require substantial preprocessing as is the case with the AMETRA corpus. In this project is shown how the statistical techniques can be succesfully applied and how the statistical and the translation memory approaches

can be combined to a translation of Spanish to Basque.

4.4 TT2: TransType2

The aim of TT2 (IST-2001-32091) is to develop a Computer-Assisted Translation (CAT) system that will help solve a very pressing social problem: how to meet the growing demand for high-quality translation. The innovative solution proposed by TT2 is to embed a data driven Machine Translation (MT) engine within an interactive translation environment. In this way, the system combines the best of two paradigms: the CAT paradigm, in which the human translator ensures high-quality output; and the MT paradigm, in which the machine ensures significant productivity gains. Another innovative feature of TT2 is that it will have two input modalities: text and speech. Six different versions of the system will be developed for English, French, Spanish and German (with English as the pivot).

Relevant results

TT2 is based on the premise that we can improve the productivity of translators by reducing the number of keystrokes needed for entering a translation. Professionals at two translation bureaus were testing the prototypes, demonstrating that TransType2 allows to increase the translator's productivity by between 15 and 20% on average.

4.5 ITEFTE:Finite State transducer inference to machine translation and machine translation assisted in specific tasks

ITEFTE project (CICYT TIC2003-08681-C02-02) aims to develop finite-state technologies for translation, such as:

a) Machine translation inference and computer assisted translation in constraint domains

b) Building translation memories of easy maintenance and fast answer

c) Producing speech to speech translation systems in constraint domains

d) Building parallel morphosyntactically-anotated corpora

e) Developing of text classifiers in order to obtain specialized translators

f) Implementation of statistical and geometric aligners enriched with linguistic information

5 Latest contributions of PRHLT in Machine Translation and Computer-Assisted Translation

In this section, a review of the most significant contributions of the PRHLT group to the field of Machine Translation and Computer-Assisted Translation is presented.

5.1 Machine Translation

The PRHLT group most important publications organized by topics are listed below:

- *Grammatical Inference*:

  J. Oncina, P. García, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Trans. on PAMI*, 15(5):448–458, 1993.

  J. M. Vilar. Improve the learning of subsequential transducers by using alignments and dictionaries. In *ICGI '00*, pages 298–311, London, UK, 2000. Springer-Verlag.

  D. Picó and F. Casacuberta. Some statistical-estimation methods for stochastic finite-state transducers. *Machine Learning*, 44:121–142, Jul.-Aug. 2001.

  E. Vidal and F. Casacuberta. Learning finite-state models for machine translation. In *Grammatical Inference: Algorithms and Applications. Proceedings of the 7th International Coloquium ICGI 2004*, volume 3264 of *LNAI*, pages 16–27. Springer, Athens, Oct. 2004. Invited conference.

  F. Casacuberta, E. Vidal, and D. Picó. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38:1431–1443, 2005.

  E. Vidal, F. Thollard, F. Casacuberta C. de la Higuera, and R. Carrasco. Probabilistic finite-state machines - part I & II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1039, 2005.

- *IBM model search/decoding algorithms*

  J. Tomás and F.Casacuberta. A statistical spanish-catalan translator: a preliminary version. In *Proceedings VIII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, pages 103–110, Bilbao, May 1999.

  I. García-Varea and F. Casacuberta. Search algorithms for statistical machine translation based on dynamic programming and pruning techinques. In *MT Summit VIII*, pages 115–120, September 2001.

D. Ortiz, I. García-Varea, and F. Casacuberta. An empirical comparison of stack-decoding algorithms for statistical machine translation. In *Pattern Recognition and Image Analysis, First Iberia Conference*, volume 2652 of *Lecture Notes in Computer Science*, pages 654–663. Springer-Verlag, Puerto de Andratx, Mallorca, June 2003.

- *Maximum Entropy*

  I. García-Varea and F. Casacuberta. Maximum entropy modeling: A suitable framework to learn context-dependent lexicon models for statistical machine translation. *Machine Learning*, 60:135–158, 2005.

- *Phrase-Based models*

  J. Tomás and F. Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the MT Summit VIII*, pages 357–361, Santiago de Compostela, 2001.

  D. Ortiz, I. García-Varea, and F. Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Tenth MT Summit*. AAMT, Phuket, Thailand, Sep. 2005.

- Translation models

  J. M. Vilar and E. Vidal. A recursive statistical translation model. In Association of Computational Linguistics, editor, *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, USA, June 2005.

  J. Andrés. N-HSEST: N-History Segmented Enumerable Stochastic Transducer. Technical Report DSIC-II/16/05, D.S.I.C., U.P.V., Nov. 2005.

5.2 Computer Assisted Translation

The PRHLT group most important publications are listed below:

Atos Origin, Instituto Tecnológico de Informática, RWTH Aachen, RALI Laboratory, Celer Soluciones and Société Gamma and Xerox Research Centre Europe. *TransType2 - Computer Assisted Translation*. Project Technical Annex., 2001.

E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-assisted translation using speech recognition. *IEEE Transaction on Speech and Audio Processing*, In press, 2005.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A.L. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.M. Vilar. Statistical and finite-state approaches to computer-assisted translation. *Computational Linguistics*, In preparation.

References

[1] P. F. Brown, J. Cocke, S. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[2] P. F. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, 1993.

[3] I. García-Varea and F. Casacuberta. Maximum entropy modeling: A suitable framework to learn context-dependent lexicon models for statistical machine translation. *Machine Learning*, 60:135–158, 2005.

[4] F.J. Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, RWTH, Aachen, 2002. Advisor: Dr. H. Ney.

[5] J. Tomás and F. Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the Machine Translation Summit VIII*, pages 357–361, Santiago de Compostela, 2001.

[6] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *EMNLP-02*, 2002.

[7] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March 1996.

[8] J.M.Vilar. *Aprendizaje de Transductores Subsecuenciales para su empleo en tareas de Dominio Restringido*. PhD thesis, U.P.V., 1998. Advisor: Dr. E.Vidal.

[9] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[10] F. Casacuberta, E. Vidal, and D. Picó. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38:1431–1443, 2005.

[11] F. Casacuberta, E. Vidal, A. Sanchis, and J.-M. Vilar. Pattern recognition approaches for speech-to-speech translation. *Cybernetic and Systems: an International Journal*, 35(1):3–17, 2004.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39(1):1–22, 1977.

[13] F. J. Och. GIZA++: Training of statistical translation models, 2000.

[14] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. pages 228–235, Toulouse, July 2001.

[15] A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. Language translation apparatus and method of using context-based translation models. U.S. Patent, No. 5510981, April 1996.

[16] C. Tillmann. *Word Re-Ordering and Dynamic Programming based Search Algorithms for Statistical Machine Translation*. PhD thesis, Computer Science Department, RWTH Aachen, Germany, May 2001.

[17] I. García-Varea. *Traducción automática estadística: modelos de traducción basados en máxima entropía y algoritmos de búsqueda*. PhD thesis, D.S.I.C., U.P.V., Dec. 2003.

[18] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March 2003.

[19] D. Ortiz, I. García-Varea, and F. Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Tenth Machine Translation Summit*, pages 141–148. Asia-Pacific Association for Machine Translation, Phuket, Thailand, September 2005.

[20] J. Tomás and F. Casacuberta. Statistical machine translation decoding using target word reordering. In *SSPR2004 and SPR2004*, volume 3138 of *Lecture Notes in Computer Science*, pages 734–743. Springer-Verlag, Lisboa, Aug. 2004.

[21] P. Koehn. *Noun Phrase Translation*. PhD thesis, Univ. of Southern California, December 2003.

[22] P. Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124, 2004.

[23] I. García-Varea, F. J. Och, H. Ney, and F. Casacuberta. Refined lexicon models for statistical machine translation using a maximum entropy approach. pages 204–211, Toulouse, July 2001.

[24] I. García-Varea, F. J. Och, H. Ney, and F. Casacuberta. Efficient integration of maximum entropy lexicon models within the training of statistical alignment models. In C. Richardson, editor, *Machine Translation: From research*

to real users, LNAI 2499, pages 161–168. Springer-Verlag, 2002. AMTA-2002 Conference. Tiburon, CA.

[25] J. Oncina, P. García, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Trans. on PAMI*, 15(5):448–458, 1993.

[26] J. Oncina and M. A. Varó. Using domain information during the learning of a subsequential transducer. In *ICGI*, pages 301–312, Berlin, 1996.

[27] J. M. Vilar. Improve the learning of subsequential transducers by using alignments and dictionaries. In *ICGI '00*, pages 298–311. Springer-Verlag, 2000.

[28] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, MA, 1998.

[29] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recoginition. *Proceedings of the IEEE*, 77:257–286, 1989.

[30] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, S. Molau, F. Nevado, M. Pastor, D. Picó, A. Sanchis, and C. Tillmann. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[31] P. Langlais, G. Foster, and G. Lapalme. Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4):267–294, 2000.

[32] J. Civera, J.M. Vilar, E. Cubel, A.L. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó, and J. González. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition*, Lecture Notes in Computer Science. Springer-Verlag, 2004.

Pattern Recognition Approaches for Speech Recognition Applications*

V. Alabau, J.M. Benedí, F. Casacuberta, A. Juan, C.D. Martínez-Hinarejos, M. Pastor, L. Rodríguez, J.A. Sánchez, A. Sanchis and E. Vidal
Pattern Recognition and Human Language Technology Research Group
Institut Tecnològic d'Informàtica
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
46071, València (Spain)

Abstract

Automatic Speech Recognition is one of the most important areas of interest in Pattern recognition. We propose a statistical approach to limited-domain speech recognition, which uses finite-state modeling at all levels. These approach is applied for different tasks of increasing interest: dialogue and computer-assisted transcription of speech. Since current speech recognition systems are not error-free, we present the use of Pattern Recognition techniques to predict the reliability of each hypothesized word. A summary of the most relevant results is reported over a wide variety of tasks, that show the real possibilities of these techniques.

Keywords: Stochastic finite-state models, speech recognition, confidence measures, computer-assisted transcription of speech, dialogue.

1 Introduction

Since the origins of Computer Science, one of the most retailing problems has been the possibility of communicating with a computer using the most natural way for a human being: speech. The achieving of this objective would spread the use of computer systems to the vast majority of human beings, who could then take advantage of all the computer applications.

The initial solutions provided to speech recognition (i.e., the mere activity of knowing the exacts words that were uttered by the speaker) were mostly based on pure Artificial Intelligence techniques (fuzzy techniques were usual). All these

* This work has been partially supported by Agencia Valenciana de Ciencia y Tecnología (GRUPOS03/031) and the Spanish project ITEFTE (TIC2003-08681-C02-02).

systems required the use of experts to build the knowledge base the recognition system was based on. As a consequence, these systems were expensive and hard to adapt to different situations.

The complementary approach is inspired in Statistical Pattern Recognition. With this approach, we try to build the speech recognition system with the minimal human intervention, getting the most information necessary for the system from available data using statistical techniques. With the increasing of available speech data in the 80s, this approach became more and more popular, and finally it overcome nearly all the other approaches.

The great success of the speech recognition systems relied on the use of statistical inference techniques to estimate the parameters of certain models. Due to the sequential nature of the speech signal, the kind of models that demonstrated more effective for this task were Finite-State Models, and more specifically, the probabilistic version of these models.

In the section 2, the Finite-State Model application to speech recognition is presented, as long as the specific kind of models which is usually used and an overview of the processes involved in the speech recognition algorithms. In section 3 a different number of speech recognition applications based on Pattern Recognition techniques are presented: speech recognition, estimation of confidence measures, computer-assisted transcription and dialogue systems.

2 Automatic Speech Recognition based on Finite-State Models

Automatic Speech Recognition (ASR) is an interesting problem that is conveniently addressed in the Pattern Recognition framework with Stochastic Finite-State models (FSM). The ASR problem can be stated from a stochastic point of view as follows [1]: suppose that Θ is a sequence of characteristic vectors that represents the acoustic signal, and $W = \langle w_1 \dots w_n \rangle$ is a sequence of n words. The probability $\Pr(W|\Theta)$ is the probability of sequence W being uttered from the acoustic sequence Θ. The ASR problem can be stated as the problem of maximizing the probability $\Pr(W|\Theta)$ as follows:

$$W^* = \arg\max_W \Pr(W|\Theta) = \arg\max_W \Pr(W)\Pr(\Theta|W) \tag{1}$$

where:

- $\Pr(W)$ is the *language model probability* and represents the probability of the word sequence;

- $\Pr(\Theta|W)$ is the *acoustic probability* and represents the probability of observing the sequence of characteristic vectors Θ when the word sequence W is uttered.

The interest of this decomposition of the ASR problem derives from the fact that there exist powerful techniques both to solve the problem of the language modeling and the acoustic modeling. We now describe these techniques.

The computation of the acoustic probability $\Pr(\Theta|W)$ is carried out by supposing that each word of the sequence $W = \langle w_1 \cdots w_n \rangle$ is composed by the concatenation of a sequence of acoustic units $D(w) = \langle u_1 \cdots u_{|w|} \rangle$. In this way, the value $\Pr(\Theta|W)$ is computed from the probabilities that are obtained for each acoustic unit of W as follows:

$$\Pr(\Theta|W) = \prod_{i=1}^{n} \prod_{j=1}^{|D(w_i)|} \Pr(o_{w_i}^{u_j}) \tag{2}$$

where $o_{w_i}^{u_j} \in \Theta$ is the acoustic sequence that is generated by the acoustic unit u_j of the word w_i.

The most successful approach that is used to represent the acoustic units is based on Hidden Markov Models (HMMs) [2, 3].

A HMM is a stochastic function of a Markov chain. A HMM is composed of two stochastic processes: on the one hand, a hidden process that represents the time evolution of the Markov chain through several states according to a transition probability function; on the other hand, an observation process according to a distribution associated to each state.

The Markov chain is represented through a stochastic FSM with two distributions: the transition probabilities and the emission probabilities (see Figure 1).

There exists robust techniques that can be used to estimate these distributions from a set of samples. The most usually technique consists of maximizing the likelihood of the sample with the Baum-Welch algorithm, that is also known as forward-backward algorithm [4, 5].

The language model of an ASR system defines a structure in the language of a task. It attaches a probability to each word sequence W and restricts the possible sequences.

Given a word sequence $W = \langle w_1 \cdots w_n \rangle$, its probability can be computed as:

$$\Pr(W) = \prod_{i=1}^{n} \Pr(w_i|w_1 \cdots w_{i-1}) \tag{3}$$

where $\Pr(w_i|w_1 \cdots w_{i-1})$ is the conditional probability of occurring w_i after the sequence $w_1 \cdots w_{i-1}$.

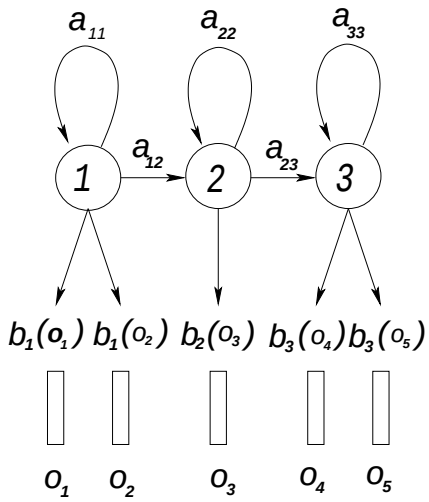


Figure 1: Example of Hidden Markov Model: a_{ij} are the transition probabilities and $b_k(o_t)$ are the emission probabilities.

The most successful approach that is currently used in ASR are n-gram models [1] (specially bigram models ($n = 2$) and trigram models ($n = 3$)). In these models, the probability of occurring a word w_i depends only on the previous $N - 1$ previous words:

$$\Pr(W) = \prod_{i=1}^{n} \Pr(w_i | w_{i-N+1} \cdots w_{i-1}) \tag{4}$$

These models can be estimated from a set of sentences. The model can be represented with a stochastic FSM (see Figure 2). In this model, each edge is labeled with a word and a probability, and the transitions between edges represent possible sequence of words. A path from an initial state to a final state represents a possible sequence of words of the language, and its probability is the product of the probabilities that appear in the path. Given that the set of samples is finite, smoothing techniques are applied in order to model unseen events [6].

Given the HMMs and the language model which can be represented like FSMs, an integrated FSM is constructed with this information. This integrated FSM is constructed by replacing each edge in the language model by the lexical model of the word associated to the edge. This lexical model represents the sequence of acoustic units of the word (see Figure 3). Then, each edge in the lexical model is substituted by the HMM corresponding to the acoustic unit (see an example of this composition in Figure 4).

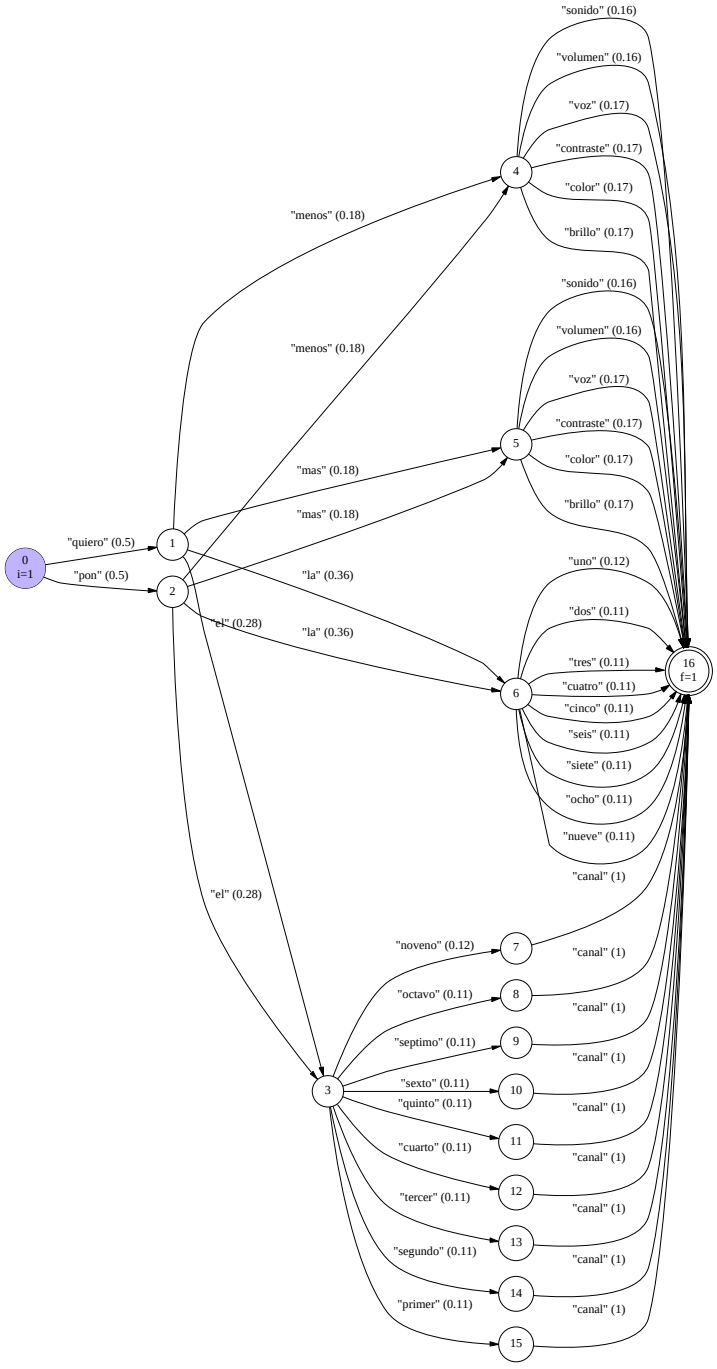


Figure 2: Stochastic FSM for modeling the speech control of a TV.

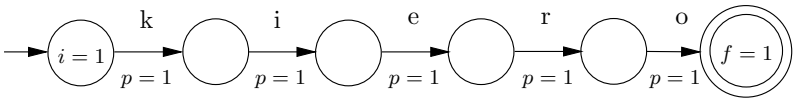
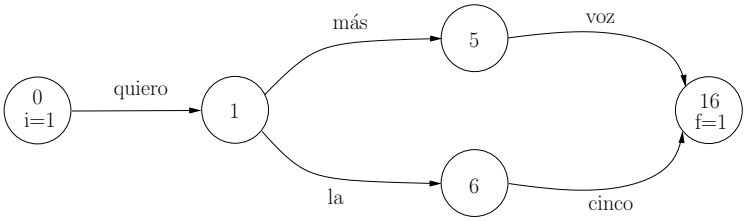
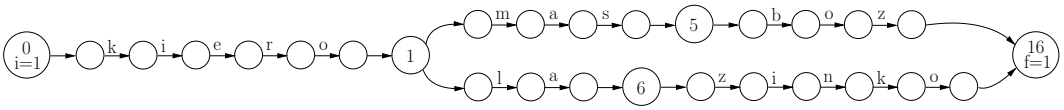


Figure 3: Example of Stochastic FSM for the lexical representation of the Spanish word "quiero".

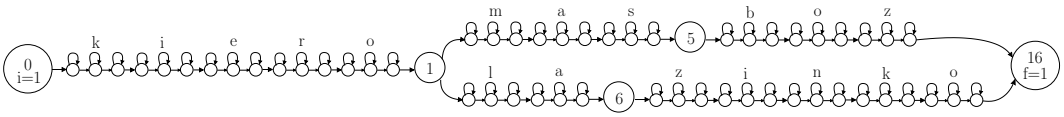
The recognition problem can be stated as a search problem in the integrated automaton. Given an acoustic sequence, the problem is to find the hypotheses W^* (sequence of states) that maximizes the probability $\Pr(W|\Theta)$. The search problem can be efficiently carried out with the Viterbi algorithm [7, 8].



(a) Language Model



(b) Lexical expansion



(c) Acoustic Expansion

Figure 4: Example of the integration process of the lexical and phonetic knowledge into an integrated FST.

3 Applications

In this section we show different speech recognition applications based on Pattern Recognition techniques: speech recognition based on FSMs, confidence estimation, computer-assisted transcription and dialogue systems.

3.1 Speech recognition

In this section we show the assessment of a speech recognizer based on FSMs through experiments with three limited-domain tasks of increasing difficulty.

ATROS (Automatically Trainable Recognizer of Speech) is a continuous-speech recognition/translation system which uses stochastic FSM at all its levels: acoustic-phonetic, lexical and syntactic/translation [9, 10]. All these models can be learn automatically from speech and text data. The use of FSMs allows the system to obtain the translation and the recognized sentences synchronously.

ATROS was developed within the framework of the European project EuTrans. The EuTrans project was aimed at developing machine translation systems to assist human to human (speech) communications in specific domains [11]. The specific domain was the translation of queries, requests and complains made by telephone (or microphone) to the front desk of a hotel.

Three tasks of different degree of difficulty were developed within the EuTrans project. In the first one (Eu-0), Spanish to English translation systems were learned from a big and well controlled training corpus (about 170k different pairs). In the second one (Eu-I), also from Spanish to English, the systems were learned from a random subset of 10k pairs from the previous corpus Eu-0, which was established as a more realistic training corpus for the kind of application considered. In the third (Eu-II), from Italian to English, the systems were learned from a small training corpus (about 3k pairs) that was obtained from a transcription of a spontaneous speech corpus. A summary of the main features of the EuTrans corpus is presented in Table 1.

The Eu-0 and Eu-I test-set speech corpus consists of telephone and microphone speech-input. The telephone-input test-set was similar except that the number of test speakers were 10 rather than 4.

3.1.1 Experimental Results

Three learning finite-state methods have been studied. The first method is a classical trigram model [1]. The second ones are OMEGA [12] and GIATI [13], which are finite-state transducers whose are used as source language models. These meth-

Table 1: Summary of the EuTrans corpus

		Eu-0	Eu-I	Eu-II
Training Text	# Sentences	$490k$	$10k$	$3k$
	# Different Sentences	$168k$	$10k$	$3k$
	Vocabulary	686		2.459
	Bigram test-set perplexity	6.8	8.6	31
Speech Test	time	$0.5h$		$0.8h$
	# Speakers	4		24
	Speech utterances	336		278
	Running words	$3k$		$5k$

ods are expected to manage cross-lingual syntactic constraints that improve ASR performance [11]. To assess the performance of the models, the (Recognition) Word Error Rate (WER) was adopted. This performance criterion, widely used in speech recognition, is basically the minimum number of substitution, insertion and deletion operations that have to be performed to convert the word string produced by a system into a given reference word string. The best results are summarized in the table 2.

Table 2: Assessment results

		WER (%)		
Speech Signal	Models	Eu-0	Eu-I	Eu-II
microphone	trigrams	2.4	4.1	-
	GIATI	2.3	4.4	-
	OMEGA	4.1	13.6	-
telephone	trigrams	8.6	11.6	22.1
	GIATI	7.5	10.5	32.0
	OMEGA	8.4	18.3	52.5

Results show that for simple tasks and using sufficient training data, all these techniques yield good results, especially for telephone speech signal. As training data shrinks, results degrade gradually. Finally, for more complex tasks, with (relatively) small amount of training data, results become generally worse. OMEGA performs badly due to inference problems.

3.2 Confidence measures for speech recognition

Current speech recognition systems are not error-free and, in consequence, it is desirable for many applications to predict the reliability of each hypothesized word. From our point of view, this can be seen as a conventional pattern recognition problem in which each hypothesized word is to be transformed into a feature vector and then classified as either correct or incorrect [14, 15]. The basic problem then is to decide which predictor (pattern) features and classification model should be used.

The problem of finding appropriate (pattern) features has been extensively studied by several authors. Some of them have noticed that correctly recognized words are often among the most probable hypotheses. Accordingly, they suggest the use of features derived from n-best lists [16, 17] or word graphs [18, 19].

To design an accurate pattern classifier, we first consider a *word-dependent* naive Bayes model in which the estimation of class posteriors is carried out using conventional relative frequencies (we assume that features are discrete). Due to the lack of training data, this model underestimates the true probabilities involving rare words and the incorrect class. To deal with this problem of data spareness, our basic model is smoothed with a generalized, *word-independent* naive Bayes model.

3.2.1 Naive Bayes model

The class variable is denoted by c; $c = 0$ for correct and $c = 1$ for incorrect. Given a hypothesized word w and a D-dimensional vector of features $\boldsymbol{x}$, the class posteriors can be calculated via the Bayes' rule as

$$P(c|\boldsymbol{x}, w) = \frac{P(c|w)\, P(\boldsymbol{x}|c, w)}{\sum_{c'} P(c'|w)\, P(\boldsymbol{x}|c', w)} \tag{5}$$

For simplicity, the model includes the naive Bayes assumption that the features are mutually independent given a class-word pair,

$$P(\boldsymbol{x}|c, w) = \prod_{d=1}^{D} P(x_d|c, w) \tag{6}$$

Therefore, the basic problem is to estimate $P(c|w)$ for each target word and $P(\boldsymbol{x}|c, w)$ for each class-word pair. Given N training samples $\{(\boldsymbol{x}_n, c_n, w_n)\}_{n=1}^{N}$, the unknown probabilities can be estimated using the conventional frequencies:

$$P(c|w) = \frac{N(c, w)}{N(w)} \tag{7}$$

$$P(x_d|c,w) = \frac{N(x_d,c,w)}{N(c,w)} \tag{8}$$

where the $N(\cdot)$ are suitably defined event counts; i.e., the events are (c,w) pairs in (7) and (x_d,c,w) triplets in (8).

In practice, some features may have continuous rather than discrete domains. In that case, the use of Eq. 8 requires the discretization of continuous features [15].

Unfortunately, these frequencies often underestimate the true probabilities involving rare words and the incorrect class. To circumvent this problem, the model is smoothed using the *absolute discounting* smoothing technique imported from statistical language modeling [20]. The idea is to discount a small constant $b \in (0,1)$ to every positive count and then distribute the gained probability mass among the null counts (unseen events). A detailed explanation of the smoothed model can be found in [21, 15].

3.2.2 Confidence estimation using Bayes decision theory

Confidence estimation can be seen as the problem of classifying new words output by the speech recognizer as either correct or incorrect, once the parameters of the chosen model have been estimated during the training stage. This is a classical two-category classification problem, with different costs associated to the two possible classification errors [22]. In our case, the loss incurred when a correct word is classified as incorrect (*false rejection*) will in general be lower than the loss incurred when an incorrect word is classified as correct (*false acceptance*). For instance, a false rejection could involve asking the user to repeat the utterance, while a false acceptance could involve giving to the user a service or response he or she did not ask for. According to [22, Secc. 2.3] minimum risk is incurred by deciding $c = 1$ *if*

$$(\lambda_{01} - \lambda_{11})P(c = 1 \mid \boldsymbol{x}, w) > (\lambda_{10} - \lambda_{00})P(c = 0 \mid \boldsymbol{x}, w) \tag{9}$$

with λ_{ij} being the loss incurred if we decide in favor of class i when the true state of nature is j. Since $P(c = 0 \mid \boldsymbol{x}, w) = 1 - P(c = 1 \mid \boldsymbol{x}, w)$, this is equivalent to deciding $c = 1$ if

$$P(c = 1 \mid \boldsymbol{x}, w) > \tau = \frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11} + \lambda_{10} - \lambda_{00}}. \tag{10}$$

Thus, the optimal decision rule consists in classifying a word as incorrect if $P(c = 1 \mid \boldsymbol{x}, w)$ is greater that a certain threshold τ that depends of the values λ_{ij}.

3.2.3 Experimental results

In evaluating confidence estimation performance, two measures are of interest: the *True Rejection Rate* (TRR, the number of incorrect words that are classified as incorrect divided by the number of incorrect words) and the *False Rejection Rate* (FRR, the number of correct words that are classified as incorrect divided by the number of correct words). The trade-off between TRR and FRR values depends on a decision threshold τ (see section 3.2.2). A *Receiver Operating Characteristic* (ROC) curve represents TRR against FRR for different values of τ. The area under a ROC curve divided by the area of a worst-case diagonal ROC curve, provides an adequate overall estimation of the classification accuracy. We denote this area ratio as AROC. Note that an AROC value of 2.0 would indicate that all words can be correctly classified. Another criterion is the *Confidence Error Rate* (CER) defined as the number of classification errors divided by the total number of recognized words. A baseline CER is obtained assuming that all recognized words are classified as correct.

As predictor features we used a number of 20 features: 12 features are based on word graphs and the other 8 were proposed by different authors [23, 16, 24]. A detailed explanation of these features can be found in [15]. Table 3 shows the best results using a subset of these features along with the naive bayes smoothed model (eq. 5). Experiments were performed using the *Eu-I* (microphone) and the *Eu-II* corpus summarized in table 1.

The use of pattern recognition techniques for confidence estimation in speech recognition achieves high performance. The naive bayes model achieves significant relative reduction in baseline CER: 27% for Eu-I and 37.8% for Eu-II. The AROC values show also a high overall estimation of the classification accuracy.

Table 3: Best results for Eu-I and Eu-II corpus

	Eu-I	Eu-II
AROC	1.86	1.81
CER	3.30	13.06
Baseline CER	4.52	21.0

3.3 Computer-Assisted Transcription of Speech

Complex tasks with large vocabularies, noisy environments, spontaneous speech, etc. result in a significant number of errors in transcriptions. When high quality transcriptions are needed, a human transcriptor is required to verify and correct the (imperfect) system's transcriptions.

This process is usually performed *off-line*. First, the system returns a full transcription of the input audio signal. Next, the human transcriptor reads it sequentially (while listening to the original audio signal) and corrects the possible mistakes made by the system. This solution is rather uncomfortable and inefficient for the human corrector.

An interactive *on-line* scenario can allow for a more efficient approach. Here, the ASR and the human transcriptor cooperate to generate the final transcription of the input signal. The rational behind this approximation is to combine the high quality provided by the human transcriptor with the efficiency of the ASR. We denote this approach as "Computer Assisted Transcription of Speech" (CATS) and it is based on the interactive approach previously applied to Computer Assisted Translation (CAT) [25, 26].

Experiments with the proposed CATS approach show that the interactive paradigm not only is more comfortable for the human transcriptor but also reduces the overall effort needed. In the next sections we show the application of pattern recognition techniques to CATS.

3.3.1 Foundations of CATS

The process starts when the ASR system proposes a full *transcription* S (or a set of best transcriptions) of a suitable segment of the acoustic signal Θ. Then, the human transcriptor (named *user* from now on) reads this transcription until he or she finds a mistake; i.e, he or she validates a prefix S_p of the transcription which is error-free. Now, the user can enter a word (or words), C, to *correct* the erroneous text that follows the validated prefix. This produces a new *prefix* P (the previously validated prefix, S_p, followed by C). Then, the ASR system takes into account the new prefix to suggest a suitable continuation (or a set of best possible continuations) to this prefix (i.e., a new S), thereby starting a new cycle. This process is repeated until a correct, full transcription of Θ is accepted by the user.

A key point on this interactive process is that, at each user-system iteration, the system can take advantage of the prefix validated so far to attempt an improved prediction for the continuation of this prefix.

The use of FSMs at all ASR levels allows to perform this process very efficiently.

3.3.2 CAT based on Pattern Recognition Framework

In section 2, we show that speech recognition is stated as the problem of searching for a sequence of words, W, that with maximum probability has produced a given utterance, Θ. In the CATS framework, in addition to the given utterance Θ, a *prefix* P of the transcription (validated and/or corrected by the user) is available and the ASR should try to complete this prefix by searching in the integrated FSM for a most likely *suffix* $\hat{s}$ as:

$$\begin{aligned} \hat{s} &= \underset{s}{\operatorname{argmax}}\, Pr(s \mid \Theta, P) \\ &= \underset{s}{\operatorname{argmax}}\, Pr(\Theta \mid s, P) \cdot Pr(s \mid P) \end{aligned} \tag{11}$$

Therefore, the search must be performed over all possible suffixes s of P and the language model probability $Pr(s \mid P)$ must account for the words that can be uttered *after the prefix* P.

In order to solve equation (11), the signal Θ can be considered split into two fragments, Θ_1^b and Θ_{b+1}^m, where m is the length of Θ. By further considering the boundary point b as a hidden variable in (11), we can write:

$$\hat{s} = \underset{s}{\operatorname{argmax}} \sum_{0 \le b \le m} \Pr(\Theta, b \,|\, s, P) \cdot \Pr(s \,|\, P) \tag{12}$$

We can now make the *naive* (but realistic) assumption that Θ_1^b do not depend on the suffix, and Θ_{b+1}^m do not depend on the prefix, and we can approximate the sum by the dominating term to rewrite (12) as:

$$\hat{s} \approx \underset{s}{\operatorname{argmax}} \max_{0 \le b \le m} \Pr(\Theta_1^b \,|\, P) \cdot \Pr(\Theta_{b+1}^m \,|\, s) \cdot \Pr(s \,|\, P) \tag{13}$$

3.3.3 Experimental results

For the experimental study, two different corpora were used. The first one is the Eu-I Corpus (described in table 1). The second is the *Albayzin geographic corpus* [27], consisting of oral queries to a geographic database.

For evaluating the CAT performance, two measures were employed: on the one hand, the Word Error Rate (WER) (described in section 3.1.1); and on the other hand, the Word Stroke Ratio (WSR) [25], a measure borrowed from CAT was employed.

Table 4 presents the results obtained with both corpora. The difference between the WER number and the WSR number indicates the reduction of effort achieved by CATS with respect to the post-edition process in *classic* ASR.

Table 4: Results obtained with the Eutrans and Albayzin corpora

	Eutrans	Albayzin
WER	11.4	11.6
WSR	9.3	10.1
% Improvement	≈ 19	≈ 14

3.4 Dialogue systems

A dialogue system is defined as a computer application that allows a user to achieve a certain objective or accomplish a defined task using dialogue. These systems are really useful in many tasks where, currently, a human operator is needed to input user queries to information systems and inform of the query results to the user. The main objective is to achieve a computer system that can simulate the human capabilities in terms of dialogue.

The application of Pattern Recognition methods and techniques to dialogue systems is not new. The first attempts in dialogue management using Finite-State technologies were produced in the 90s. In these proposals, the dialogue space was defined in terms of a FSM, where each state is associated to the state of the data required by the dialogue system [28, 29]. The behavior of the system was based on transiting from one state to another depending on the user input, and performing a different action depending on the state the system was. The problem with this approach was the combinatorial explosion of states even for a limited number of data items [28].

A new step was taken to use probabilistic dialogue models, which define the dialogue actions in terms of Dialogue Acts (DA) [30]. An DA defines which is the intention and mission of the utterance at dialogue level (e.g., if it is a question, an answer, what about it is, etc.). Thus, these models try to associate a sequence of DA to a given user input, and try to define the actions to be taken by the system as a sequence of new DA. The model features can be learn automatically from dialogues annotated with the corresponding DA.

The initial works were directed to the labelling of dialogue turns from models derived from a Maximum Likelihood approach. This approximation results in using models like HMMs and n-grams [31, 32]. In this framework, it is supposed we have available the sequence of words W that constitute the dialogue segment (a.k.a. utterance) and the previous sequence of the l assigned DA D in the current dialogue. Using this information, the DA to assign to the segment D^* is defined by:

$$D^* = \underset{D'}{\operatorname{argmax}} \Pr(D'|W) \Pr(D \cdot D') \approx \underset{D'}{\operatorname{argmax}} \Pr(D'|W) \Pr(D'|D^l_{l-n})$$

Clearly, $\Pr(D'|W)$ can be defined by a HMM and $\Pr(D'|D^l_{l-n})$ is an n-gram model, given a method very similar to isolated word recognition. Using this model, high accuracy results (approximately 71%) are achieved for the SwitchBoard task [32].

Although these models are initially defined for text input, some works were done to include other features, like prosody, which are exclusively from speech [32]. In these works, the $\Pr(D'|W)$ is substituted by the adequate distribution probability that includes all the other acoustic features which are taken into account.

More recently, some work has been done in the initial direction of defining the system behavior [33], based on the same type of models. In this case, apart from extending the assignation of DA to whole unsegmented user turns, the model presents the reaction of the system in terms of DA sequences. Thus, the reaction of the system corresponds to a (sequence of) DA D^* which is given by:

$$D^* = \underset{\mathcal{D}}{\operatorname{argmax}} \left[\max_{D} \Pr(\mathcal{D}|{d'}^s_{s+2-m}) \prod_{i=1}^{l} \Pr(D_i|{D'}^{i-1}_{i+1-n}) \Pr(\Omega_i|D_i) \right]$$

In this formula, it is assumed a certain segmentation of the user turn words Ω in l segments (Ω_i) for the sake of simplicity, but the real process performs simultaneously both segmentation and assignation, in a very similar manner to continuous speech recognition. The used models are again HMM for $\Pr(\Omega_i|D_i)$, and n-gram models for $\Pr(D_i|{D'}^{i-1}_{i+1-n})$ and $\Pr(\mathcal{D}|{d'}^s_{s+2-m})$.

Many works have been developed in the line of mixing classical Pattern Recognition and Reinforcement Learning techniques, using the framework of the Markov Decission Processes (MDP). An MDP is defined as a tuple of states, actions, transitions and a reward function, and the probabilistic distributions that drive its behavior can be inferred with the classical Estimation-Maximisation algorithms [34, 35]. As an extension of MDP, Partially Observable MDP have been proposed recently for dialogue management [36].

The application of Pattern Recognition based techniques on dialogue is not only limited to dialogue management. One of the most important tasks in dialogue systems development is the annotation of the data used to infer the probabilistic models. This annotation task is usually manual and very expensive. Therefore, applications devoted exclusively to annotate dialogue corpora have been developed, some of them using FSMs derived from Grammar Inference techniques [37]. The results achieved

with these models for Basurde task (about 47% of the turns completely well labeled) show that they are really appropriate to save time in the dialogue labelling task.

In the same line, other works were directed to the identification of the DA of the turns using the combination of HMM and n-grams [32], or using the presence in the turn of significant n-grams (cue-phrases) to determine the most likely DA [38]. All these approximations use classical statistical classification techniques, which are frequent in Pattern Recognition.

3.5 Concluding remarks

In this paper we have shown the use of Pattern Recognition techniques for different speech recognition applications. For speech recognition we have presented an approach which use stochastic FSMs at all its levels: acoustic-phonetic, lexical and syntactic/translation. Good recognition results were achieved for three tasks of different degree of difficulty. For confidence estimation, we have presented a sound framework based on Bayes decision theory. We propose a smoothed naive bayes model for estimating confidence measures. High performance has been achieved using this model along with a set of well-known features. A new approach to the production of perfect transcriptions of speech has been presented. This approach combines the efficiency of an ASR system with the accuracy of a human transcriptor. The results are very promising even with this initial approximation. Finally, we have presented a brief review of the pattern recognition techniques applied to dialogue systems.

References

[1] F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, 1998.

[2] J.K. Baker. The dragon system - an overview. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1(23):143–159, 1975.

[3] F. Jelinek. Continuous speech recognition by statistical methods. *Proc. IEEE*, 4(64):532–556, 1976.

[4] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[5] P.A. Devijver. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3:369–373, 1985.

[6] H. Ney, U. Essen, and R. Knesser. On structuring probabilistic dependences in stochastic language modelling. *Computer, Speech and Language*, 8:1–38, 1994.

[7] A. Viterbi. Error bounds for convolutional codes and an asymtotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[8] G.D. Forney. The viterbi algorithm. In *Proceedings of the IEEE*, volume 61(3), pages 268–278, 1973.

[9] D. Llorens, F. Casacuberta, E. Segarra, J.A. Sánchez, and P. Aibar. Acoustical and syntactical modeling in atros system. In *International Conference on Acoustic, Speech and Signal Processing*, volume 2, pages 641–644. IEEE Press, March 1999.

[10] F. Casacuberta, D. Llorens, C. Martínez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, and J. M. Vilar. Speech-to-speech translation based on finite-state transducers. In *International Conference on Acoustic, Speech and Signal Processing*, volume 1. IEEE Press, April 2001.

[11] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, C. Martinez D. Llorens, S. Molau, F. Nevado, M. Pastor, D. Pico, and A. Sanchis. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[12] J.M.Vilar. *Aprendizaje de Transductores Subsecuenciales para su empleo en tareas de Dominio Restringido*. PhD thesis, Universidad Politécnica de Valencia, 1998. Advisor: Dr. E.Vidal.

[13] D. Picó, J. Tomás, and F. Casacuberta. GIATI: A general methodology for finite-state translation using alignments. In *Statistical, Structural and Syntactical Pattern Recognition. Proceedings of the Joint IAPR Intternational Workshops SSPR2004 and SPR2004*, volume 3138 of *Lecture Notes in Computer Science*, pages 216–223. Springer-Verlag,, Lisboa, Portugal, August 2004.

[14] A. Sanchis, V. Jiménez, and E. Vidal. Efficient use of the grammar scale factor to classify incorrect words in speech recognition verification. In *International Conference on Pattern Recognition ICPR-2000*, volume 3, pages 278–281, Barcelona, Spain, September 2000.

[15] A. Sanchis. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. PhD thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, May 2004.

[16] L. Chase. *Error-responsive feedback mechanisms for speech recognizers*. PhD thesis, School of Computer Science, Carnegie Mellon University, USA, 1997.

[17] T.J. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Recognition confidence scoring for use in speech understanding systems. *Computer Speech and Language*, 16(1):49–67, 2002.

[18] T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *European Conf. on Speech Technology, (EUROSPEECH)*, pages 827–830, 1997.

[19] F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

[20] H. Ney, S. Martin, and F. Wessel. Statistical language modeling using leaving-one-out. *Young, S. and Bloothoft, G., editors, Corpus Based Methods in Language and Speech Processing*, pages 174–207, 1997.

[21] A. Sanchis, A. Juan, and E. Vidal. Improving utterance verification using a smoothed naive bayes model. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, 2003.

[22] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1974.

[23] T. Zepenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel. Recognition of conversational telephone speech using the JANUS speech engine. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, pages 1815–1818, 1997.

[24] A. Sanchis, A. Juan, and E. Vidal. Estimating confidence measures for speech recognition verification using a smoothed naive bayes model. In Francisco José Perales, Aurélio J. C. Campilho, Nicolás Pérez de la Blanca, and Alberto Sanfeliu, editors, *Pattern Recognition and Image Analysis, First Iberian Conference IbPRIA 2003 Proceedings*, Lecture Notes in Computer Science LNCS 2652, pages 910–918, Port d'Andratx, Mallorca, Spain, jun 2003. Springer-Verlag.

[25] E. Cubel, J. Civera, J. M. Vilar, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, J. González, and L. Rodríguez. Finite-state models for computer assisted translation. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI04)*, pages 586–590, Valencia, Spain, 2004.

[26] J. Civera, J.M. Vilar, E. Cubel, A.L. Lagarda, F. Casacuberta, E. Vidal, D. Picó, and J. González. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition – Joint IAPR Imternational workshops on Syntactical and Structural Pattern Recognition (SSPR 2004) and Statistical Pattern Recognition (SPR 2004)*, Lecture Notes in Computer Science. Springer-Verlag, Lisbon, Portugal, August 2004.

[27] J.E. Díaz-Verdejo, A.M. Peinado, A.J. Rubio, E. Segarra, N. Prieto, and F. Casacuberta. Albayzin: a task oriented spanish speech corpus. In *Proceedings of First Intern. Conf. on Language Resources and Evaluation (LREC-98)*, volume 1, pages 497–501, 1998.

[28] N. M. Fraser, B. Salmon, and T. Thomas. Call routing by name recognition: Field trial results for the operetta(tm) system. In *IVTTA'96*, NJ, USA, 1996.

[29] M. McTear. Modelling spoken dialogues with state transition diagrams: experiences of the cslu toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1223–1226, Sydney, Australia, 1998. Australian Speech Science and Technology Association, Incorporated.

[30] J. R. Searle. *Speech acts*. Cambridge University Press, 1969.

[31] T. Fukada, D. Koll, A. Waibel, and K. Tanigaki. Probabilistic dialogue act extraction for concept based multilingual translation systems. In *Proceedings of International Conference on Spoken Language Processing*, volume 6, pages 2771–2774, 1998.

[32] A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer. Dialogue act modelling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):1–34, 2000.

[33] C. D. Martínez-Hinarejos and F. Casacuberta. Evaluating a probabilistic dialogue model for a railway information task. In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Proceedings of the Fifth International Conference on Text, Speech and Dialogue—TSD 2002*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2448, pages 381–388, Brno, Czech Republic, Sep 2002. Springer-Verlag.

[34] E. Levin, R. Pieraccini, and W. Eckert. Using markov decision processes for learning dialogue strategies. In *Proceedings of International Conference on Speech, Signal and Audio Processing (ICASSP)*, Seattle, WA, May 1998.

[35] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, January 2000.

[36] J. D. Williams, P. Poupart, and S. Young. Partially observable markov decision processes with continuous observations for dialogue management. In *Proceedings of the 6th SigDial Workshop on Discourse and Dialogue*, Lisbon, September 2005.

[37] C. D. Martínez-Hinarejos and F. Casacuberta. A pattern recognition approach to dialog labelling by using finite-state transducers. In *Proceedings of 5th. IberoAmerican Symposium on Pattern Recognition*, pages 669–677, Lisbon, Portugal, September 2000.

[38] N. Webb, M. Hepple, and Y. Wilks. Dialogue act classification using intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, Pittsburgh, 2005.

Classifier ensembles for genre recognition *

Pedro J. Ponce de León, José M. Iñesta, Carlos Pérez-Sancho
Universidad de Alicante.
Departamento de Lenguajes y Sistemas Informáticos
P.O. box 99, E-03080 Alicante, Spain
{*pierre,inesta,cperez*}*@dlsi.ua.es*
`http://grfia.dlsi.ua.es`

Abstract

Previous work done in genre recognition and characterization from symbolic sources (monophonic melodies extracted from MIDI files) have pointed our research to the use of classifier ensembles to better accomplish the task. This work presents current research in the use of voting ensembles of classifiers trained on statistical description models of melodies, in order to improve both the accuracy and robustness of single classifier systems in the genre recognition task. Different voting schemes are discussed and compared, and results for a corpus of Jazz and Classical music pieces are presented and assesed.

Keywords: Statistical pattern recognition, Classifier ensembles, Music information retrieval, Musical genre recognition

1 Introduction

Some recent works explore the capabilities of machine learning or pattern recognition methods to recognise music genre, either using audio [1, 2, 3], or symbolic [3, 4, 5] sources, or even metadata [6]. After a period of time doing research on the use of statistical models and classification paradigms for music genre (or style) characterization from symbolic data [7, 8], we reached a point where the combination of the different learning systems we developed showed up as the logical next step in our research. The many ways of building classifier *ensembles* (i.e., combining different classifiers) to improve both the accuracy and robustness of single classifiers is a hot topic in the areas of machine learning or pattern recognition. Works on this subject point out the importance of the concept of *diversity* in classifier ensembles, with respect to both classifier outputs and structure [9, 10, 11].

Our current research on combination of several previously developed classification systems for genre recognition in the symbolic domain is presented in this paper.

*This work was supported by the projects Spanish CICyT TIC2003–08496–C04, partially supported by EU ERDF, and Generalitat Valenciana GV043–541.

MIDI files have been used as the primary source of music data so, first, the music corpus of such files used is described. Second, the statistical description models utilized to describe music content are presented. Next, the classification techniques based on them are described, along with the different ensemble schemes for combining classifier decisions. Following this, the results for the ensembles are presented and compared with individual classifier results for genre recognition. Finally, the conclusions drawn from the results are discussed, pointing the research to further work lines.

2 Music data

The music corpus used is a set of MIDI files from *Jazz* and *Classical* music with a monophonic melody track, collected from different sources. No preprocessing of these files was done before entering the system, except for manually checking the presence and correctness of key, tempo, and meter meta-events, as well as the labeling of the melody track.

The corpus is made up of 110 files. 45 files are classical music files and 65 are jazz files, with a total length around 10,000 bars (more than six hours of music). The classification systems presented here work only on the information contained in the melody track. The rest of the MIDI file content is ignored because one of the general aims of this work is to analyze how much of the genre information is contained in the melody alone.

Two different ways of describing the content of the melody track have been used. The first one is based on melodic, harmonic, and rhythmic statistical descriptors and the second one describes melodic content in terms of strings of symbols corresponding to melody subsequences. Both description methods are briefly described in the following sections.

3 Statistical description models

3.1 Shallow structure descriptors

The first group of description models that have been used are based on descriptive statistics that summarise the content of a melody in terms of pitches, note durations, silences, harmonicity, rhythm, etc. This kind of statistical description of musical content is sometimes referred to as *shallow structure description* [12].

In these models, each melody is described by a vector of statistical descriptors, labeled with the genre of the melody. A set of 28 descriptors has been defined,

based on several categories of features that assess melodic, harmonic, and rhythmic properties of a melody. These descriptors are summarized in Table 3.1. The first column indicates the musical property analysed and the other columns indicate the kind of statistics describing the property. A blank entry in the table means that a particular statistic has not been computed.

Four different description models have been defined. The model containing all the descriptors is called the F (full) model. From this one, three reduced models have been derived. This has been achieved using a per-feature separability test described in [13] to rank the features. Subsets of features are incrementally built by choosing the best ranked features. These models are called here A, B, and C for simplicity. Model A includes the six best ranked features, model B adds four features to model A, and model C adds two features to model B, so that $A \subset B \subset C \subset F$. Each entry in Table 3.1 indicates the smallest feature subset where the particular statistical descriptor has been included.

Category	Counter	Range	Avg. (relative)	Dev.	Normality
Notes	A				
Significant silences	B				
Non significant silences	F				
Pitches		A	A	A	F
Note durations		F	F	C	F
Silence durations		F	F	F	F
Inter-onset intervals		F	F	B	F
Pitch intervals		A	F	B	B
Non-diatonic notes	F		F	C	F
Syncopations	A				

Table 1: Shallow structure descriptors

For the descriptor computations, the melodies are quantized to a resolution of $Q = 48$ ticks per bar. Durations are measured in ticks. For pitch and interval categories, the range descriptors are computed as the maximum minus the minimum value in the melody, and the average-relative descriptors are computed as the average value minus the minimum value. For durations (note and silence durations, and inter-onset intervals) the range descriptors are computed as the ratio between the maximum and the minimum values, and the average-relative descriptors are computed as the ratio between the average and the minimum value. Finally, normality descriptors are computed using the D'Agostino statistic [14] for assessing the normality of the distribution of each property.

3.2 n-word based descriptors

The n-word based models make use of text categorization methods to describe melodic content. The technique encodes note sequences as character strings, therefore converting a melody in a text to be categorized. Such a sequence of n consecutive notes is called an n-word. All possible n-words in a melody are extracted, except those containing a silence lasting four or more beats. The encoding for n-words used in this work has been derived from the method proposed in [15]. This method generates n-words by encoding pitch interval and duration information. For each n-note sequence, all pitch intervals and duration ratios (inter-onset interval ratio) are calculated using Eqs. (1) and (2) respectively:

$$I_i = Pitch_{i+1} - Pitch_i \qquad (i = 1, \ldots, n-1) \tag{1}$$

$$R_i = \frac{Onset_{i+2} - Onset_{i+1}}{Onset_{i+1} - Onset_i} \qquad (i = 1, \ldots, n-2) \tag{2}$$

and each n-word is defined as a string of symbols:

$$[\ I_1\ \ R_1\ \ \ldots\ \ I_{n-2}\ \ R_{n-2}\ \ I_{n-1}\ \ R_{n-1}\ ] \tag{3}$$

where the pitch intervals and duration ratios have been mapped into alphanumeric characters (see [8, 15] for details).

This method represents a musical piece as a vector $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{i|\mathcal{V}|}\right)$, where each component represents the presence of the word w_t in the melody, being $|\mathcal{V}|$ the size of the vocabulary, that is, the total number of different n-words extracted from the corpus.

A common practice in text classification is to reduce the dimensionality of those vectors (usually very high) by selecting the words that contribute most to discriminate the class of a document (a melody here). The *average mutual information* measure (AMI) [16] has been used in this work to rank the words. This measure gives a high value to those words that appear often in melodies of one genre and are seldom found in melodies of the other genres. The n-words are sorted using this value, so only information about the first $|\mathcal{V}|$ words are provided to the classifier.

4 Classification techniques

4.1 Classifiers for shallow statistical features

Two different classification paradigms have been used with the four description models presented in section 3.1: the k-nearest-neighbour classifier, and the bayesian

classifier assuming non-diagonal covariance matrices [17]. For the first one, given a sample $\mathbf{x}_i$, the distances to the prototypes in the training set are computed, and the class labels of the closest k are taken into account to take the decision by a majority. A value $k = 7$ has been establish for this classifier after some trials.

In the bayesian classifier the classification is performed following the well-known *Bayes' classification rule*. In a context where there is a set of classes $c_j \in \mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$, a sample $\mathbf{x}_i$ is assigned to class c_j with maximum a posteriori probability, in order to minimize the probability of error:

$$P(c_j|x_i) = \frac{P(c_j)P(x_i|c_j)}{P(x_i)} \quad . \tag{4}$$

Using these two different classification techniques, eight different classifiers have been defined using the four shallow structure description models presented in section 3.1. Each classifier has been trained separately on the musical corpus and its accuracy estimated through leave-one-out cross-validation.

4.2 Naive Bayes classifier for n-words

For n-word based melody categorization, the naive Bayes classifier, as described in [18], has been used. Here, the classifier is based on the same Eq. 4, but applying the *naive Bayes assumption*, i.e. it is assumed that all words in a melody sample are independent of each other, and also independent of the order they were generated. This assumption is clearly false in our problem and also in the case of text classification, but naive Bayes can obtain near optimal classification errors in spite of that [19].

In this work, classes are musical genres, and the class-conditional probability of a melody $P(\mathbf{x_i}|\mathbf{c_j})$ is given by the probability distribution of note sequences (n-words) in genre c_j, which can be learned from a labeled training set, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. Two different distribution models have been used for the class-conditional probability: a Multivariate Bernoulli (MB) model, where the components of a sample vector are $x_{it} \in \{0, 1\}$ and a Multinomial (MN) model, where components are $x_{it} \in \{0, 1, ..., |\mathbf{x}_i|\}$, being $|\mathbf{x}_i|$ the number of n-words extracted from melody $\mathbf{x_i}$. Both MB and MN distributions have proven to achieve quite good results in text classification [18] and are briefly described below.

In the MB model, each class follows a multivariate Bernoulli distribution where the parameters to be learned from the training set are the class-conditional probability of each word in the vocabulary.

The MN model takes into account word frequencies in each melody, rather than just the occurrence or non-occurence of words, as in the MB model. In consequence,

each component x_{it} is the number of occurrences of word w_t in the melody. In this model, the probability that a melody has been generated from a genre c_j is a multivariate multinomial distribution, where the melody length is assumed to be class-independent [18].

4.3 Classifier ensembles

After analysing the performance of the different classifiers studied, we have found a diversity of errors among the decisions taken by the different classifiers. This diversity has been suggested by some authors [10, 20] as an argument for using classifier ensembles with good results. These ensembles could be regarded as committees of 'experts' [21] in which the decisions of individual classifiers are considered as opinions supported by a measure of confidence usually related to the accuracy of each classifier. The final classification decision is taken either by majority vote or by a weighing system.

4.3.1 Voting schemes.

Designing a suitable method of decision combination is a key point for the ensemble's performance. In this paper, different possibilities that are presented below have been explored and compared. In the discussion that follows, N stands for the number of samples contained in the training set $\mathcal{X} = \{\mathbf{x}\}_{i=1}^{N}$, M is the number of classes in a set $\mathcal{C} = \{c_j\}_{j=1}^{M}$, and K classifiers, C_k, are utilized.

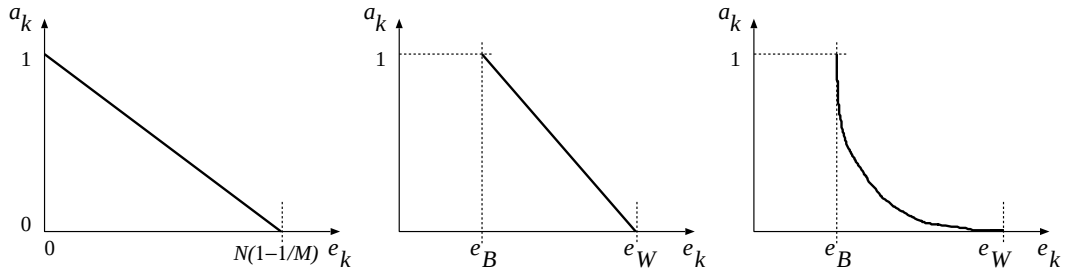


Figure 1: Different models for giving the authority (a_k) to each classifier in the ensemble as a function of the number of errors (e_k) made on the training set.

1. Majority vote. This is the simplest method. It just counts the number of decisions for each class and assigns the sample $\mathbf{x}_i$ to the class c_j that obtained the

highest number of votes. The snag here is that all the classifiers have the same 'authority' regardless of their respective abilities to classify properly. In terms of weights it can be considered that $w_k = 1/K \ \forall k$.

2. Simple weighted majority. The decision of each classifier, C_k, is weighed according to its estimated accuracy (the ratio of successful classifications, α_k) on the training set [22]. This way, the authority for C_k is just $a_k = \alpha_k$. Then, its weight w_k is:

$$w_k = \frac{a_k}{\sum_l a_l} \quad . \tag{5}$$

Also for the rest of weighting schemes presented here, the weights are the normalized values for a_k, as shown in this equation.

The weak point of this scheme is that an accuracy of 0.5 in a two-class problem still has a fair weight although the classifier is actually unable to predict anything useful. This scheme has been used in other works [23] where the number of classes is rather high. In those conditions this drawback may not be evident.

3. Re-scaled weighted majority. The idea is to assign a cero weight to classifiers that only give N/M or less correct decisions on the training set, and scale the weight values proportionally, assigning $a_k = 1$ to the perfect classifier. As a consequence, classifiers with an estimated accuracy $\alpha_k \leq 1/M$ are actually removed from the ensemble. The values for the authority are computed according to the line displayed in figure 1-left. Thus, if e_k is the number of errors made by C_k, then

$$a_k = \max\{0, 1 - \frac{M \cdot e_k}{N \cdot (M-1)}\} \quad .$$

4. Best-worst weighted majority. In this ensemble, the best and the worst classifiers in the ensemble are identified using their estimated accuracy. A maximum authority, $a_k = 1$, is assigned to the former and a null one, $a_k = 0$, to the latter, being equivalent to remove this classifier from the ensemble. The rest of classifiers are rated linearly between these extremes (see figure 1-center). The values for a_k are calculated as follows:

$$a_k = 1 - \frac{e_k - e_B}{e_W - e_B} \quad ,$$

where $e_B = \min_k\{e_k\}$ and $e_W = \max_k\{e_k\}$.

5. Quadratic best-worst weighted majority. In order to give more authority to the opinions given by the most accurate classifiers, the values obtained by the former approach are squared (see figure 1-right). This way,

$$a_k = (\frac{e_W - e_k}{e_W - e_B})^2 \quad .$$

4.3.2 Classification.

Once the weights for each classifier have been computed, the class receiving the highest score in the votation is the final class prediction. If $\hat{c}_k(\mathbf{x}_i)$ is the prediction of C_k for the sample $\mathbf{x}_i$, then the prediction of the ensemble can be computed as

$$\hat{c}(\mathbf{x}_i) = \arg\max_{c_j \in \mathcal{C}} \sum_k w_k \delta(\hat{c}_k(\mathbf{x}_i), c_j) \quad , \tag{6}$$

being $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

Since the weights represent the normalized authority of each classifier, it follows that $\sum_{k=1}^{M} w_k = 1$. This makes possible to interpret the sum in Eq. 6 as $P(c_j|\mathbf{x}_i)$, the probability that $\mathbf{x}_i$ is classified into c_j, and $\hat{c}(\mathbf{x}_i)$ as the class for which this probability is maximum.

5 Results

The classifiers described in sections 4.1 and 4.2 have been utilized in order to build the ensembles, combining the different description models and classification paradigms: four k-nearest neighbors, using $k = 7$, with the different feature combinations (A, B, C, and F models), four Bayesian classifiers with the same feature combinations, and two naive Bayes using Bernoulli and Multinomial probability distributions. For the latter, a vocabulary size of 100 and 170 2-words have been used respectively, according to their AMI values. This makes a total of ten classifiers for building ensembles. Table 5 presents the estimated accuracy of the individual classifiers, α_k, obtained using a leave-one-out validation method on the training set.

Five different ensembles have been constructed using the five different votation methods described above (represented here as V1, V2, V3, V4, and V5). The decisions of the ensembles are summarised in Table 5 (*# errors all* column), and graphically depicted in Fig. 2 against the best individual classifier score. Note that the ensemble's performance using the quadratic best-worst strategy improves the behaviour of the best of the individual classifiers: just two errors against the three

Classification paradigm	Statistical model	Feature selection	# errors	α_k
7-nearest neighbours	Shallow	A	7	0.936
	Shallow	B	12	0.891
	Shallow	C	12	0.891
	Shallow	F	3	0.973
Bayes	Shallow	A	10	0.909
	Shallow	B	9	0.918
	Shallow	C	10	0.909
	Shallow	F	22	0.746
Naive Bayes	Bernoulli	$\|\mathcal{V}\| = 100$	8	0.923
	Multinomial	$\|\mathcal{V}\| = 170$	16	0.855

Table 2: Working parameters and accuracy of the different classifiers selected.

Voting method	# errors all	%	# errors all-but-best	%
V1	6	94.5	5	95.5
V2	9	91.8	9	91.8
V3	5	95.5	8	89.1
V4	3	97.3	4	96.4
V5	2	98.2	4	96.4

Table 3: Ensemble's performance.

errors made by 7-nearest neighbour classifier based on the whole set of shallow descriptors. Also it is interesting to see that majority voting and simple or re-scaled weighted majority perform clearly worse than the best-worst scale-based schemes.

The question arises of how sensitive is this success to the construction of the ensemble. In addition, is it worth to build an ensemble for avoiding just one error? The answer for both questions could be approached removing from the ensemble the best of the classifiers and analysing how much the performance is degraded. Thus, the 7-nearest neighbour classifier trained with the F model was dropped from the ensemble, and the new results were those also shown in Table 5 (*# errors all-but-best* column).

Note how, although the results are not as good as earlier, some ensembles mantain a high standard of precision, with just 4 errors. This clearly improves the performance of the current best classifier (7 errors), so the ensemble seems quite robust and performs well, specially with the best-worst strategies introduced here.

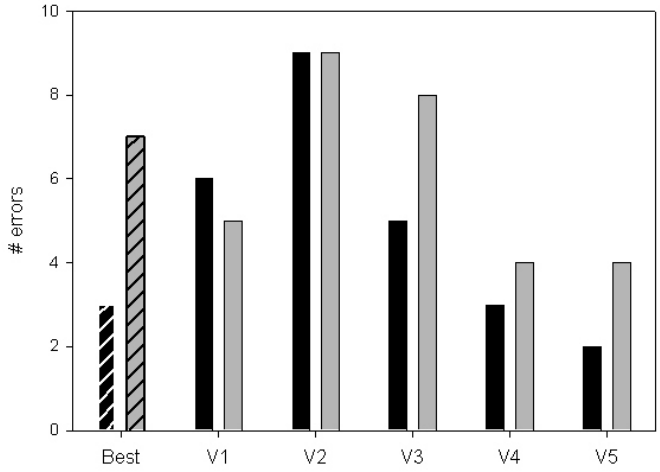


Figure 2: Number of errors made by the different ensembles (voting schemes from 1 to 5, and the performance of the best classifier on the left). Bars in black correspond to the ensemble of all the classifiers and grey bars to the ensemble of all but the best.

6 Conclusions

We have shown the performance of classifier ensembles for classifying a symbolically represented melody into a given music genre, using statistical description models. In previous works we have shown the feasibility of using these kind of data and representations to approach the problem, but by constructing an ensemble using different classifiers, their votes are "averaged" and this reduces the risk of choosing the wrong classifier.

Among all the voting schemes tested, the approaches based on scaling the weights to a range established by the best and worst classifiers have shown the best classification accuracy, which is slightly better than the most accurate individual classifier utilized. Evidence of the robustness of these best-worst scale based ensembles has also been shown. After removing the best classifier from the ensembles, they still managed to perform fairly better than any of the remaining individual classifiers.

Further work is needed to test the robustness of this scheme to other music genres, using different classification paradigms, and combination techniques, perhaps taking advantage of the capability of the combination schemes presented here to

ouput membership probabilities for each genre, given a sample melody, as stated in section 4.3.2.

7 Acknowledgments

The authors would like to thank Francisco Moreno-Seco for his help, advise, and programming.

References

[1] Jianjun Zhu, Xiangyang Xue, and Hong Lu. Musical genre classification by instrumental features. In *Int. Computer Music Conference, ICMC 2004*, pages 580–583, 2004.

[2] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568, 2001.

[3] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, June 2003.

[4] P. P. Cruz, E. Vidal, and J. C. Pérez-Cortes. Musical style identification using grammatical inference: The encoding problem. In Alberto Sanfeliu and José Ruiz-Shulcloper, editors, *Proc. of CIARP 2003*, pages 375–382, La Habana, Cuba, 2003.

[5] Cory McKay and Ichiro Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Int. Conf. on Music Information Retrieval, ISMIR 2004*, pages 525–530, 2004.

[6] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist classification with web-based data. In *Proceedings of the 5th International ISMIR 2004 Conference*, Barcelona, Spain, October 2004.

[7] Pedro J. Ponce de León and José M. Iñesta. Statistical description models for melody analysis and characterization. In *Proceedings of the 2004 International Computer Music Conference*, pages 149–156. International Computer Music Association, 2004.

[8] C. Pérez-Sancho, J. M. Iñesta, and J. Calera-Rubio. Style recognition through statistical event models. In *Proceedings of the Sound and Music Computing Conference, SMC '04*, 2004.

[9] T. Dietterich. Ensemble methods in machine learning. In *First Internacional Workshop on Multiple Classifier Systems*, pages 1–15. 2000.

[10] L. I. Kuncheva. That elusive diversity in classifier ensembles. In *Proc. 1st Iberian Conf. on Pattern Recognition and Image Analysis (IbPRIA'03)*, volume 2652 of *Lecture Notes in Computer Science*, pages 1126–1138. 2003.

[11] Derek Partridge and Niall Griffith. Multiple classifier systems: Software engineered, automatically modular leading to a taxonomic overview. *Pattern Analysis and Applications*, 5:180–188, 2002.

[12] Jeremy Pickens. A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts, 2001.

[13] P. J. Ponce de León and J. M. Iñesta. Feature-driven recognition of music styles. In *1st Iberian Conference on Pattern Recognition and Image Analysis. LNCS, 2652*, pages 773–781, 2003.

[14] R. B. D'Agostino and M. A. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, Inc., New York, 1986.

[15] Shyamala Doraisamy and Stefan Rüger. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70, 2003.

[16] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[17] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2000.

[18] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[19] P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of simple bayesian classifier. *Machine Learning*, 29:103–130, 1997.

[20] Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *Machine Learning: ECML 2000, 11th*

European Conference on Machine Learning, volume 1810 of *Lecture Notes in Computer Science*, pages 109–116. 2000.

[21] A. Blum. Empirical support for winnow and weighted-majority based algorithms: Results on a calendar scheduling domain. *Machine Learning*, 26(1):5–23, 1997.

[22] D. Opitz and J. Shavlik. Generating accurate and diverse members of a neural-network ensemble. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 535–541, 1996.

[23] E. Stamatatos and G. Widmer. Music performer recognition using an ensemble of simple classifiers. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 335–339, 2002.

An edit distance for ordered vector sets with application to character recognition*

Juan Ramón Rico-Juan†, José Manuel Iñesta†§
† Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos.
E-03071 Alicante, Spain

Abstract

In this paper a new algorithm to describe a binary image as an ordered vector set is presented. An extension of the string edit distance is defined for computing it between a pair of ordered sets of vectors. This edit distance can be used in nearest neighbor classification tasks. The advantages of this method applied to isolated handwritten character classification are shown, compared to similar methods based in string or tree representations of the binary image.

1 Introduction

The description of an object contour in a binary image as a string [1] using Freeman code [2] or using a tree representation structure [3, 1] is widely used in pattern recognition. For using these structures in a recognition task, the edit distance is often used as a measure of the differences between two prototypes. Both, string edit distances [4] and tree edit distance [5] are used, depending on the data structures utilized for representing the problem data. In this paper, to obtain a representation of the object contour from a binary image, an ordered vector set is extracted, and an edit distance measure is defined between pairs of instances of this representation. This distance is an extension of the string edit distance, adding two new rules and changing vectors by symbols. The goal is to reduce the features that represent a binary image in order to compute the distance faster, keeping the final classification time low and good error rates.

2 Feature extraction from a binary image

The goal of this representation is to describe the contour of an object using the least possible number of elements. The classical representation of a contour in a binary image links the pixels with their neighbors using 0 to 7 (see Fig. 1) codes which

*Work partially supported by the Spanish CICYT under contract TIC2003-08496-CO4

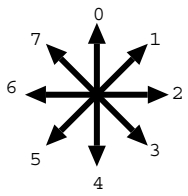


Figure 1: Freeman 2D code

represent a discrete number of 2D directions. This way, a chain that represents the contour is obtained (Fig. 2 top-left).

This kind of feature extraction assumes that all linked pixels are of equal importance. If we select the most representative points of the contour and link all these ppints, a compact representation of 2D figures is obtained, with less features than using Freeman codes.

The idea is to select a set of dominant points in a contour [6, 7], link those dominant points following the contour of the figure using 2D vectors, and then use these ordered vector set to represent the image (Fig. 2 bottom-right).

In a particular application of handwritten character recognition, it is recommended to apply some filter operations to original image before extracting and coding the contours [8] including an opening filter [9] and a thinning algorithm [10] in order to remove noise and redundant information.

3 Ordered vector set edit distance

The string edit distance definition [4] is based on three edit operations: insertion, deletion, and substitution. Let Σ the alphabet, $A, B \in \Sigma^*$ a finite string of characters and Λ is a null character. $A\langle i\rangle$ is the ith character of the string A; $A\langle i:j\rangle$ is the substring form the ith to jth characters of A, both inclusive.

An edit operation is a pair $a, b \in \Sigma \cup \{\Lambda\} : (a, b) \neq (\Lambda, \Lambda)$. So, the basic edit operations are substitution $a \to b$, insertion $\Lambda \to b$ and deletion $a \to \Lambda$. If a generic cost function is associated to each operation $\gamma_s(a \to b)$, the cost of the sequence of edit operations that transforms a finite string A in B is defined as

$$d_s(A,B) = \min \begin{cases} \gamma_s(\Lambda \to B\langle 1\rangle) + d_s(A, B\langle 2:|B|\rangle) & |B| > 1 \\ \gamma_s(A\langle 1\rangle \to \Lambda) + d_s(A\langle 2:|A|\rangle, B) & |A| > 1 \\ \gamma_s(A\langle 1\rangle \to B\langle 1\rangle) + d_s(A\langle 2:|A|\rangle, B\langle 2:|B|\rangle) & |A| > 1 \ \wedge \ |B| > 1 \\ 0 & |A| = 0 \ \wedge \ |B| = 0 \end{cases}$$

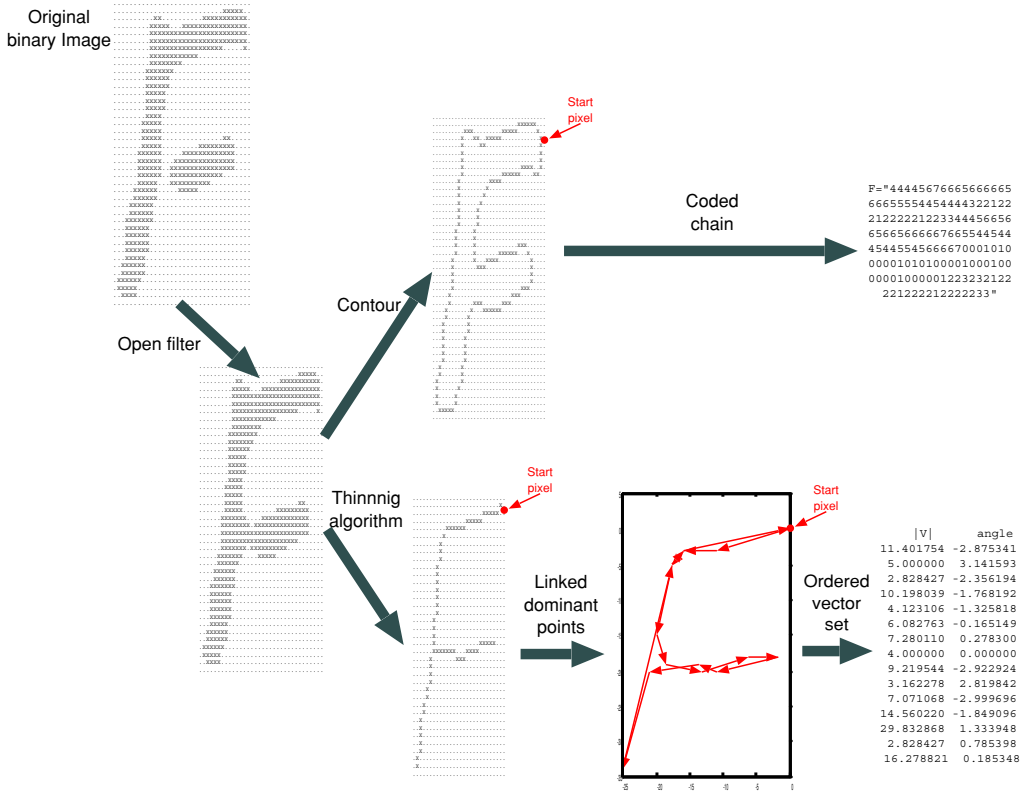


Figure 2: General scheme. From the binary image, morphological filters are applied to correct gaps and both contour and skeleton are obtained. From the first, the chain code is obtained and from the second, the ordered vector set is extracted using a dominant point selection algorithm.

The similar idea of an ordered string is extended to an ordered vector set. Let $V, W \in (\mathbb{R}\times[0,2\pi])^*$ a finite set of vectors and Λ is a null vector. $V\langle i\rangle$ is the vector ith in the set V, $V_N\langle i\rangle$ is the norm and $V_\alpha\langle i\rangle$ is the angle of ith vector; $V\langle i:j\rangle$ is the subset form ith to jth component vectors of V, both included.

Now, an edit operation is a pair $(v,w) \in (\mathbb{R}\times[0,2\pi])\,, (v,w) \neq (\Lambda,\Lambda) : (v,w^*) \cup (v^*,w)$. So, the basic edit operations are substitution (1 to 1) $v \to w$, substitution (1 to N) called fragmentation $v \to w^+$, substitution (N to 1) called consolidation $v^+ \to w$, insertion $\Lambda \to w$ and deletion $v \to \Lambda$. In this case we considered the case that one vector could be replaced by N, or vice versa.

When using dominant points, it is usual that a small change in the contour generates a new dominant point, so when comparing two prototypes 1 vector in the first prototype can be similar to N continuous vectors from the second prototype.

The cost of sequence of edit operations that transforms a finite string V into W, if we associate a cost function $\gamma_v(v^*,w^*)$, is defined as

$$d_v(V,W) = \min \begin{cases} \gamma_v(\Lambda \to W\langle 1\rangle) + d_v(V, W\langle 2:|W|\rangle) & |W|>1 \\ \gamma_v(V\langle 1\rangle \to \Lambda) + d_v(V\langle 2:|V|\rangle, W) & |V|>1 \\ \gamma_v(V\langle 1\rangle \to W\langle 1\rangle) + d_v(V\langle 2:|A|\rangle, W\langle 2:|B|\rangle) & |V|>1 \wedge |W|>1 \\ \underset{j\in[2,|W|]}{\gamma_v(V\langle 1\rangle \to W\langle 1:j\rangle) + d_v(V\langle 2:|V|\rangle, B\langle j+1:|W|\rangle)} & |W|>2 \\ \underset{i\in[2,|V|]}{\gamma_v(V\langle 1:i\rangle \to W\langle 1\rangle) + d_v(V\langle j+1:|V|\rangle, B\langle 2:|W|\rangle)} & |V|>2 \\ 0 & |V|=0 \wedge |W|=0 \end{cases}$$

The algorithm proposed in [4] for computing the string edit distance can be extended to compute the ordered vector set edit distance in the following way:

```
1. Function vectorEditDistance(V,W)
2.    D[0,0] := 0;
3.    for i := 1 to |V| do D[i,0] := D[i-1,0] + γv(V⟨i⟩ → Λ);
4.    for j := 1 to |W| do D[0,j] := D[0,j-1] + γv(Λ → W⟨j⟩);
5.    for i := 1 to |V| do
6.       for j := 1 to |W| do
7.          m1 := D[i-1,j-1] + γv(V⟨i⟩ → W⟨j⟩);
8.          m2 := D[i-1,j] + γv(V⟨i⟩ → Λ);
```

```
9.        m_3 := D[i, j-1] + γ_v(Λ → W⟨j⟩);
10.       m := ∞;
11.       for k := 1 to |V| do
12.         if (i-k) ≥ 0 then m := min{m, D[i-k, j-1] + γ_v(V⟨i-k : i⟩ → W⟨j⟩)};
13.       endfor
14.       for k := 1 to |W| do
15.         if (j-k) ≥ 0 then m := min{m, D[i-1, j-k] + γ_v(V⟨i⟩ → W⟨j-k : j⟩)};
16.       endfor
17.       D[i, j] := min(m, m_1, m_2, m_3);
18.     endfor
19.   endfor
20. return D[i, j]
```

The complexity of the string edit distance algorithm is proportional to the length of both strings, $\mathcal{O}(|A|\,|B|)$. In the case of the *vectorEditDistance*, it has a three nested loops and the complexity is $\mathcal{O}(|V|\,|W|\max\{|V|\,|W|\})$, but if we considered that a vector could be replaced by a fixed constant number of vectors, the new complexity is $\mathcal{O}(|V|\,|W|)$. Thus, the cost is similar to string edit distance.

To compute the difference between one vector and a set of N vectors, used in *vectorEditDistance*, the following function is utilized:

```
1. Function γ_v(V⟨k⟩ → W⟨i : j⟩)
2.   float auxN := 0, aunAng := 0, r := 0, rSubs := 0, rLeft := 0
3.   auxN := V_N⟨k⟩  //Norm single vector
4.   auxAng := V_α⟨k⟩  //Angle single vector
5.   for l := i to j do
6.     if auxN ≥ 0 then //Left norm single vector
7.       rSubs := rSubs + auxN * closest(auxAng, W_α⟨l⟩)
```

8. $auxAng := W_\alpha \langle l \rangle$
9. endif
10. $auxN := auxN - W_N \langle l \rangle$
11. endfor
12. if $auxN \geq 0$ then //Left norm single vector
13. $rLeft := auxN *$ kInsertion
14. else //Norms W vectors $> V$
15. $rLeft := -auxN *$ kDeletion
16. endif
17. return $rSubs + rLeft$

where closest$(angle1, angle2)$ returns the smallest angle between both parameters, resulting a value in $[0, \pi]$. The kInsertion = kDeletion $= \pi/2$ is the maximum possible difference between two angles.

The functions $\gamma_v (V \langle i.j \rangle \rightarrow W \langle k \rangle)$ and $\gamma_v (V \langle i \rangle \rightarrow W \langle j \rangle)$ are similar. In the first case, the parameters change the order and in the second case, both parameters are unitary vectors.

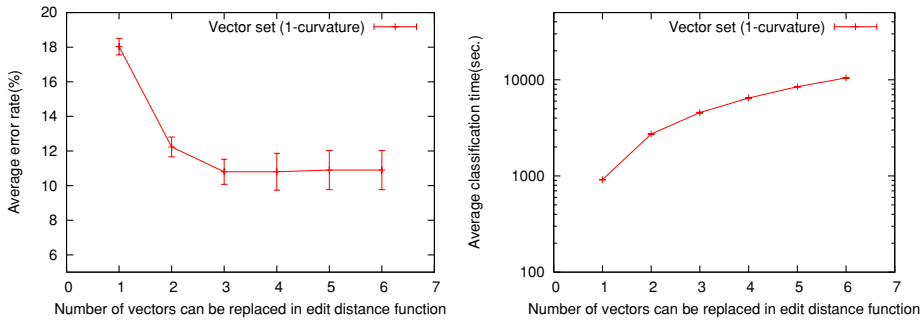
The insertion and deletion functions are defined as $\gamma_v (\Lambda \rightarrow W \langle j \rangle) = |W \langle j \rangle| *$ kInsertion and $\gamma_v (V \langle i \rangle \rightarrow \Lambda) = |V \langle j \rangle| *$ kDeletion.

4 Experiments

Three algorithms have been compared based in different contour descriptions:

1. Classical Freeman chain code extracted from the object contour in the binary image.
2. The ordered vector set extracted from the dominant points described in [7], that will be referred as non collinear dominant points (NCDP).
3. The new structure based in the ordered vector set extracted from dominant points described in [6]. In this article, $1-$curvature and $k-$curvature algorithms are defined. The authors showed that the obtained dominant points were similar for both methods, so we utilized the faster one: $1-$curvature.

In the preliminary trials tested, the algorithm $1-$curvature obtained lower error rates than NCDP. Thus, the k parameter in the *vectorEditDistance* function was tuned when applied to $1-$curvature. The k parameter is the maximum number of continuous vectors that was set to 1.



(a) (b)

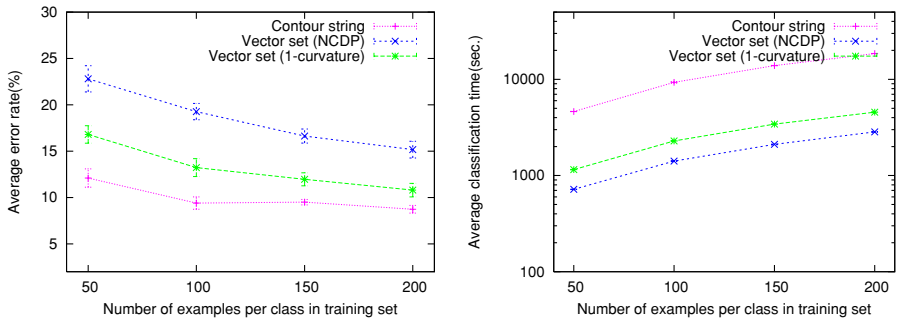
Figure 3: Results for NN classification of character obtained with ordered vector set ($1-$curvature), different training set (200 examples per class) and test set (50 samples per class and 26 character classes) as a function of different number of vectors that can be replaced in a substitution operation in a vector edit distance: (a) average error rate $\pm$ standard deviation; (b) average classification time.

A classification task using the NIST SPECIAL DATABASE 3 of the National Institute of Standards and Technology was performed using the different contour descriptions enumerated above to represent the characters. Only the 26 uppercase handwritten characters were used. The increasing-size training samples for the experiments were built by taking 500 writers and selecting the samples randomly. The nearest neighbor (NN) technique was used to perform classification.

Figure 3 shows the comparison between the error rate in the vector classification task evaluated for different sizes, k (*vectorEditDistance*). This experiment shows that the error rate decreases linearly when the k grows to a limit while increasing the number of computations increases the time of the classification. In this case, we choice the lowest error rate with the lowest k, so the optimal parameter value was $k = 3$.

The figure 4 shows the classification error rate and the time used in the classification of 50 examples per class as a function of different training set.

In all cases the use of Freeman chain codes generates a lower error rate (less than 9%) in recognition than using ordered vector sets, although the classification time



(a) (b)

Figure 4: Results for NN classification of characters obtained with different contour representations as a function of different training example sizes: (a) average error rate ± standard deviation; (b) average classification time.

is much higher. Thus, the ordered vector set description based on dominant points $1-$curvature [6] is a good trade-off choice. It obtains also a low error rate (less than 11%) and it is 10 times faster than using the Freeman chain codes.

5 Conclusions and future work

The ordered vector set that represents the contour of an object in a binary image (based in dominant points computing using 1-curvature is one order of magnitude faster than using Freeman chain codes, and it has just a slightly higher error rate. The edit distance defined in this paper to compare ordered vector sets has a similar complexity than string edit distance. Since the size of ordered vector set is significatively lower than that of strings for representing the same object, the time needed for computing the distance needed for classification is much lower.

As it can be seen in the results section the error rate using ordered vector set based based on dominant points is similar to that of using the Freeman chain code.

As future work we planned to use some especial labels in each vector to describe the curved shape of the original image to obtain a better description of the binary image contour and decrease the error rate in this classification task.

References

[1] Rico-Juan, J.R., Micó, L.: Comparison of AESA and LAESA search algorithms using string and tree edit distances. Pattern Recognition Letters **24(9)** (2003) 1427–1436

[2] Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computer **10** (1961) 260–268

[3] Rico-Juan, J.R., Micó, L.: Some results about the use of tree/string edit distances in a nearest neighbour classification task. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: Pattern Recognition and Image Analysis. Number 2652 in Lecture Notes in Computer Science, Puerto Andratx, Mallorca, Spain, Springer (2003) 821–828

[4] Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM **21** (1974) 168–173

[5] Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal of Computing **18** (1989) 1245–1262

[6] Teh, C.H., Chin, R.T.: On the detection of dominant points on digital curves. IEEE Trans. Pattern Anal. Mach. Intell. **11** (1989) 859–872

[7] Iñesta, J.M., Buendía, M., Sarti, M.A.: Reliable polygonal approximations of imaged read objects though dominant point detection. Pattern Recognition **31** (1998) 685–697

[8] Rico-Juan, J.R., Calera-Rubio, J.: Evaluation of handwritten character recognizers using tree-edit-distance and fast nearest neighbour search. In Iñesta, J.M., Micó, L., eds.: Pattern Recognition in Information Systems, Alicante (Spain), ICEIS PRESS (2002) 326–335

[9] Serra, J.: Image Analysis and mathematical morphology. Academic Press (1982)

[10] Carrasco, R.C., Forcada, M.L.: A note on the Nagendraprasad-Wang-Gupta thinning algorithm. Pattern Recognition Letters **16** (1995) 539–541

Biometric security applications *

J. García-Hernández, R. Paredes, J.C. Pérez Cortés
J. Cano, I. Salvador, E. Vidal and F. Casacuberta
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
{jgarcia,rparedes,jcperez,jcano,issalig,evidal,fcn}@iti.upv.es

Abstract

Biometric automatic identification has become an important issue in our days, because there are a large number of systems that need it in a networked society. Biometrics takes advantage of a number of unique, reliable a stable personal physiological features, to offer an effective approach to identify subjects. These features can be: iris, fingerprints, palmprints, hand geometry, face, voice, retina, hand veined pattern, etc. However, for different reasons, only some of them can be used in real systems. In this work we present the current main research areas on biometric identification of the *Instituto Tecnológico de Informática*. More particularly we present the biometric identification by: fingerprints, voice, face, palmprint and fusion methods.

Keywords: biometrics, identification, security, control access, physiological features, behavioural features.

1 Introduction

Biometric identification methods [11, 12, 2] are those that allow us to recognise a subject using physiological or behavioural features. Although the methods need that the subject must be present in the identification place, the subject collaboration is not needed in some cases and even the subject could be unaware of the system existence.

On the one hand, the *physiological methods* (figure 1) are based on the recognition of different physiological features: fingerprints, iris, retina, hand geometry, palmprint, face, DNA, hand veins pattern, face heat distribution, etc. On the other hand the *behaviour based methods* are be based on the recognition of behavioural

*Work supported by the "Agencia Valenciana de Ciencia y Tecnología (AVCiT)" under grant GRUPOS03/031 and the Spanish Project DPI2004-08279-C02-02

features: speaker identification, hand write identification, typewriter analysis, step analysis, etc.

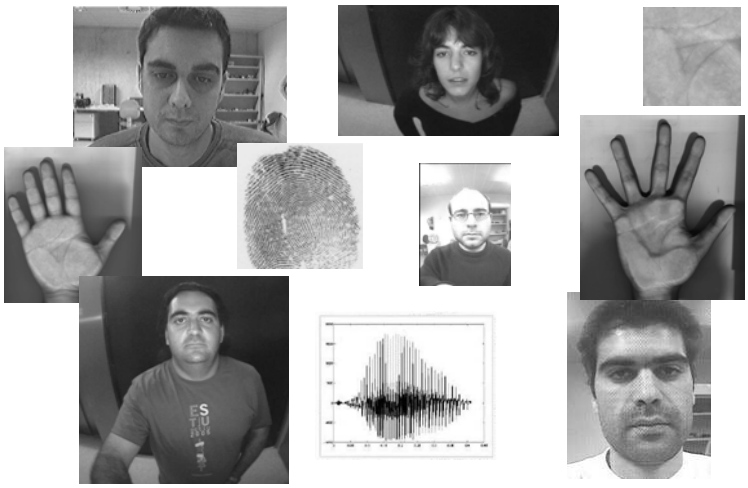


Figure 1: Biometric identification methods.

There are two main applications of a biometric identification system: *verification* and *identification*. In the first case, the subject is identified by a non-biometric method, for instance a pin code or an identification card, and the system has to verify if the given identity is correct. In the second case, the goal is to find the subject identity among a set of possible identities included in a database with biometric patterns.

For instance, typical verification applications are: building access control, computer system access control, identity control, identification in voting, use of services (cash dispenser, public carrying, etc), services payment (e-commerce), forensic identification (corpse, fatherhood, etc). A number of identification application can also be named: forensic fingerprint identification, detection of subject included in "watch lists" subject detection in public places (terrorism, crime, etc), frontiers control, etc

In systems based on pin or password identification the system performance is based on the confidentiality of the pin or password and, if a key or identification card are used, in avoiding its lost or stole. In all cases the key or code introduction provides a successful access to the system. However, in biometric system, due to the variability of the processed information it can result in a false rejection of an authorised subject or a false acceptance of an unauthorised subject.

In practise, an intermediate solution between user comfort (each false rejects is

followed by a new access try or an unnecessary alarm) and system security is suggested. The *False Accept Rate (FAR)* is the probability of access of an unauthorised subject and the *False Reject Rate (FRR)* indicates how often authorised subjects are rejected and they must repeat the identification process. FAR must be fairly small, in a range from 0.0001% to 0.1%. For instance, in USA nuclear plants, hand geometry readers with a FAR of 0.1% are used. Must be considered that the real FAR is obtained by the multiplication of the FAR by the probability for an unauthorised subject to access to the identification device and try the access. If the biometric system is joined with a classical access method, for instance magnetic card or pin, the intruder must also have the card, a copy of it or must know the pin. The FRR should also be small to avoid the authorised users' disappointment. For instance, in a device with 1000 access per day and a FRR of 1% there are 10 incidences per day.

The validation of manufacturer's rates is no easy to check due to the small percentages used. A test of thousands of supervised accesses are needed to obtain significant results from a statistical point of view.

Next we present in detail the most used biometric features and their main advantages and drawbacks.

2 Fingerprints

Fingerprint identification [8] (figure 2) is the oldest [1] of the useful biometric methods and it is widely used. The fingerprint is obtained by finger ink-impression on paper or in other material because of the flows perspired by the skin, or by the finger exploration using a electronic device. The goal is to obtain the crests distribution presented in fingertips.

Figure 2: Fingerprints

These crests make a complex pattern that is considered unique for each subject. In twins, patterns are similar but not equal. There is scientific evidence of the

low probability of that two fingerprint from two different finger could be equals by random.

Traditionally, obtained features from fingerprints have been: type and minutiaes. On the one hand, fingerprints can be classified in different type and subtype using different methods and taxonomies resulting an easier fingerprints search. On the other hand the minutiaes are crests bifurcations and endings whose relative positions can identify the fingerprints, join with the center position and structures called deltas. In a typical fingerprint we can find between 50 and 100 minutiaes.

In order to obtain these minutiae, a fingerprint preprocessing is done (figure 3). The preprocessing step filters the original image and binarises and slims the crests avoiding as possible the influence of spots, small scares and wastes that can be present in the acquisition moment.

Figure 3: Fingerprints preprocessing

Besides minutiaes comparison, there are other automatic comparison methods between fingerprints. Those methods use the correlation of crest images previously preprocessed or their directions detected by filters. The methods could achieve a good performance. However, they have problems due to the elastic deformation of a given finger. This problem causes that the methods are not efficient for searching in great finger sets.

The main advantages of fingerprints identification are:

- High universality. Any finger or hand absence is not usual.
- High permanence. Is known that finger lines do not change along subjects life.
- High oneness. Is very unprobable that the fingerprint of two different finger were identical.
- Good performance. There are efficient algorithms for matching fingerprints. The minutiaes basic information can be saved in a small storage space.

- High acceptability. This identification method is used since many years ago, so subjects see it as an usual method. However, in some cases is can be associated with criminality and private invasion.

Its main drawbacks are:

- Simplicity of measurement. Even though electronic scanners have become very cheap and have an easy installation and support, a good sample acquisition is always determined by dirty, scars, injuries, etc. A great number of subjects don't known how to place correctly the finger in the scanner.
- Although the method has a good acceptability, some subjects don't like to touch a sensor touched by many people.

3 Speaker identification

Speak [10] (figure 4) is one of the features we use to identify subjects and, in daily life, allow us to easily identify them. It is a natural way of interaction with the environment and so pronounce words or phrases to a microphone for identifying is very acceptable for users.

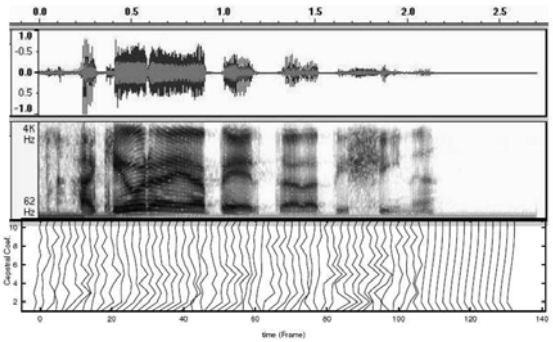


Figure 4: Speaker identification

The specific features of each subject's speak are due to differences in physiological and behaviour features of the speak system. The shape of vocal tract (larynx, pharynx, oral cavity, nasal cavity, etc) has the main roll because modifies severely the spectrum of the generated wave.

The great variation of the voice of a given subject along relatively short periods of time, and the moderate specificity of obtained features, make that speaker recognition methods are often used joined with other identification method, as intelligent card, pin, etc

The more significant advantages of speaker recognition are:

- Easy measurement: the price of the needed hardware (microphones) is small and the acquisition is very simple and comfortable for users.
- High universality: the are very few persons with voice diseases.
- Good performance: nowadays, the verification is possible with normal computing resources and the research is also possible for saved sets with a small or medium storage size. The size of saving data can be easily stored in today systems.
- High acceptability: most users do not worry for saying a word or phrase to get access to buildings or services.

Its main disadvantages are:

- Low permanence: basic voice parameters can be modified easily due to a great number of factors in short time intervals.
- Low oneness: the ability to distinguish two subjects is only medium even for humans, because an important similarity of vocals parameters is not strange.
- Low fraud detection: a high quality recorded voice would allow the access if the phrase to pronounce is not, for instance, variable or random.

4 Face identification

Face identification [3, 4, 13, 9, 7] (figure 5) is a very active pattern recognition area with a wide range of application and is one of the methods with a bigger growing. A face recognition system has to deal with variation of face images in viewpoint, illumination, background, gesture and facial details. It is a complex and very interesting problem because it has many applications. Unfortunately, it generates important distrusts in subjects, mainly in people worried with possible outrage against privacy and people's rights by technology.

Nowadays it is an active research area, so there is not agreement with the best features and comparison methods. Anyway, the idea is to save local data (eyes,

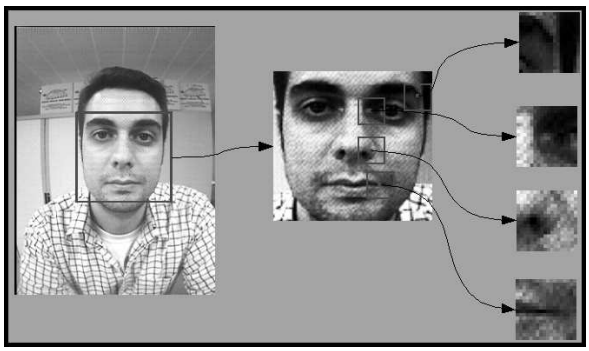


Figure 5: Face identification

mouth, nose, etc) and global data (position in the face of each feature) and join them in a model that allow us identification and an efficient search.

A typical system has two main steps. In the first one the idea is to find the face in the image, recognising it from background. In the second one the face is preprocessed and its parameters are compared with those previously saved. From the flexibility of the first step depends the system range of application and from the precision of the second one, its performance.

As in hand geometry or speaker identification, in this moment face identification can not be used in security applications with a great sets of suspects or high security access requirements by itself. It must be used joined with classical methods, as cards or pins. The last attempts of use it in terrorist localisation have been a great failure. The most known of them is the Florida Police Department attempt in Tampa airport.

On the one hand, its main advantages are:

- Easy measurement. The price of the needed hardware (cameras) is small and images acquisition can be unknown for subjects.
- High universality: each face can be found if it is not hidden by clothes.
- Good performance: Verification is possible with normal computing resources and the research is also possible for saved sets with a small or medium size (in the range of a few hundreds of faces). The size of the saved data can be stored in current systems easily.
- High acceptability: users are not delayed in its access or work.

On the other hand, its main drawbacks are:

- Low permanence: face appearance can quickly and easily change by using beard, glasses, hair, etc.
- Low oneness: nowadays, the ability to distinguish two subjects is only medium.
- Low fraud detection: the use of disguises or accessories as glasses, hats, shawls, make-up, dyeing and even haircuts or specific hairstyles can confuse the system. Other fraud ways as masks or photographs are possible, but they are difficult to use if 3D systems of thermal images are used.

5 Palmprint identification

The set of palm hand lines [5] (figure 6), from the beginning of fingers to the wrist, is other feature that can allow us to identify subjects [6]. Although years ago the prints of inked palms on paper were used, nowadays cameras and scanners are used in palm acquisition.

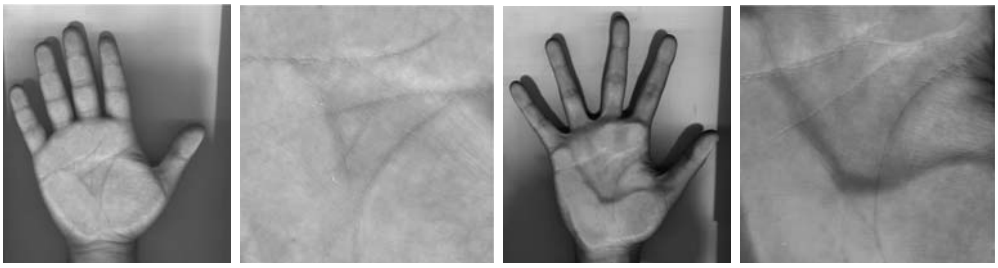


Figure 6: Palmprint identification

Principal lines are the main palmprint feature. However, a big number of secondary lines are also important. These secondary lines are called wrinkles and they look like the finger minutiaes. The set composed by these two sets is considered an unique pattern for subjects identification. As in the fingerprints case, twin brothers can be also distinguished and their palmprints are similar, but not identical. Moreover, hand lines are more difficult to blind than finger minutiaes by dirty or acidics.

The main advantages of this method are:

- High measurement simplicity. Hardware (cameras and scanners) is cheap. High resolution or colour images are not required.

- Good performance. There are efficient algorithms for palmprint verification. The palmprint basic information can be saved in a small storage space.
- High oneness. Is very unprovable that the palmprint of two different hand could be identical.
- High permanence. It is known that the essential invariance of palmprint remains along the life of subjects.
- It is also known that the palmprint pattern is more difficult to hide than fingerprint patterns by using dirty or acidics.
- The method has a good acceptability on subjects.

The main disadvantages are:

- If a scanner is used as input device many subjects do not like touching a device touched by other subjects.
- The acquisition can not be unknown for users.
- This method is newer than others, thus it is not so well-known as them.

6 Biometric security system from the "Instituto de Tecnología Informática"

The *Instituto de Tecnología Informática* (ITI) has a long experience in the development of biometric security systems. The first one was an AFIS system (Automatic Fingerprints Identification System).

There are many biometric systems, each one with its field of application. However, today the trend in high security system is to use more than one approximation at the same time [15, 14]. This *multiway approximation* (figure 7) reduces FAR and FRR, and improves the system performance.

In our days, ITI efforts are focused in face recognition and speaker recognition. Both of them provide high acceptability and have a low price of input devices. In both system users do not have neither to touch nor to interface directly with the input device. Users are only required to place in front of the camera and speak. One of the main drawbacks of the system is the low fraud detection.

Due to this drawback, ITI is now working in joining both biometric identification systems. The composition of both systems gives a better confidence reliability than the obtained by the isolated systems. The face recognition result join with the

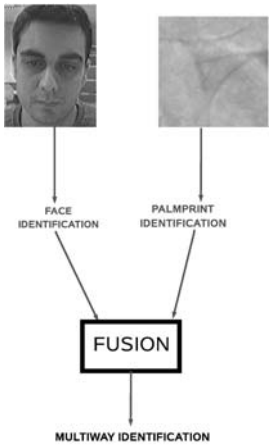


Figure 7: Multiway biometric identification

speaker recognition result are joined in an unique value that the system evaluates to compute the final result.

Nowadays the ITI has developed several prototypes and demonstrations for these technologies of biometric security. First, an access control system has been developed to allow or refuse the access to buildings, using both technologies: face and speaker recognition (figure 8). Second, other entry-system that uses palmprint recognition (figure 9) has been developed. Third, a detection and face recognition demo is available. Finally, a set of development libraries and tools (SDK) have been developed for integrating this technology in a given application.

References

[1] J. C. Amengual, A. Juan, J. C. Pérez, F. Prat, S. Sáez, and J. M. Vilar. *Real-Time Minutiae Extraction in Fingerprint Images.* In Proc. of the 6th Int. Conf. on Image Processing and its Applications (IPA 97), pages 871-875, Dublin, July 1997

[2] Ruud Bolle and Sharath Pankanti *Biometrics, Personal Identification in Networked Society*, Editor Anil K. Jain, Publisher Kluwer Academic Publishers, 1998

[3] R. Paredes, J. C. Pérez, A. Juan, and E. Vidal. *Local Representations and a direct Voting Scheme for Face Recognition.* In Proc. of the Workshop on Pattern

Figure 8: Face and speaker recognition access control.

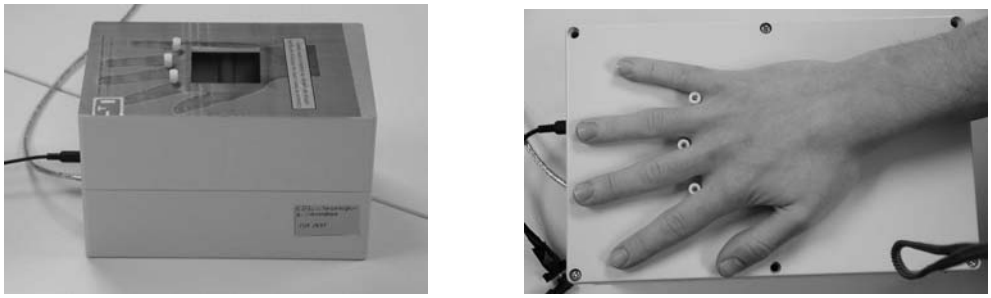


Figure 9: Palmprint entry-system.

Recognition in Information Systems (PRIS 01), Setbal (Portugal), July 2001.

[4] R. Paredes, J.C. Pérez, A. Juan and E. Vidal *Face Recognition using Local Representations and a direct Voting Scheme* Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis, pages 249-254, Benicssim (Spain), may 2001

[5] David Zhang, Wai-Kin Kong, Jane You and Michael Wong. *Online palmprint identification.* IEEE Transaction on Pattern Analysis and Machine Learning, 25(9):1041–1050, September 2003.

[6] José García-Hernández and Roberto Paredes. *Biometric identification using palmprint local features*, In Procs of 3nd COST 275 Workshop. Biometrics on the Internet Fundamentals, Advances and Applications, pages 11-14, 2005.

[7] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld *Face recognition: A literature survey*, ACM Computing Surveys (CSUR), Volume 35, Issue 4, Pages: 399 - 458, (December 2003)

[8] Anil K. Jain and David Maltoni *Handbook of Fingerprint Recognition* Publisher Springer-Verlag New York, Inc. 2003

[9] Kieron Messer, Josef Kittler, Mohammad Sadeghi, Sebastien Marcel, Christine Marcel, Samy Bengio, F.Cardinaux, C.Sanderson, J. Czyz, L. Vandendorpe, Sanun Srisuk, Maria Petrou, Werasak Kurutach, Alexander Kadyrov, Roberto Paredes, B. Kepenekci, F.B. Tek, G. B. Akar, Farzin Deravi, and Nick Mavity. *Face verification competition on the xm2vts database.* In 4th International Conference on Audio and Video Based Biometric Person Authentication, pages 964-974, June 2003.

[10] R. Paredes, E. Vidal, and F. Casacuberta. *Local features for speaker recognition.* In SPR 2004.International Workshop on Statistical Pattern Recognition. LNCS 3138 of Lecture Notes in Computer Science, pages 1087-1095, 2004.

[11] Salil Prabhakar, Sharath Pankanti and Anil K. Jain *Biometric Recognition: Security and Privacy Concerns*, IEEE Security and Privacy archive, Volume 1, Issue 2 (March 2003), Pages: 33 - 42

[12] Kresimir Delac and Mislav Grgic *A Survey of Biometric Recognition Methods* 46th International Symposium Electronics in Marine, ELMAR-2004, 16-18 June 2004, Zadar, Croatia

[13] Stan Z. Li and Anil K. Jain (Editors) *Handbook of Face Recognition* Editorial Springer (2004)

[14] Anil K. Jain and Arun Ross *Multibiometric systems* Communications of the ACM archive, Volume 47 , Issue 1 (January 2004), SPECIAL ISSUE: Multimodal interfaces that flex, adapt, and persist table of contents, Pages: 34 - 40 , 2004

[15] Arun Ross and Anil Jain, *Information fusion in biometrics*, Pattern Recognition Letters, v.24 n.13, p.2115-2125, September 2003

Hyperspectral Kernel Classifiers*

Gustavo Camps-Valls†, Luis Gomez-Chova†, Javier Calpe-Maravilla†,
Jordi Muñoz-Marí†, José D. Martín-Guerrero†,
Luis Alonso-Chordá‡, and José Moreno‡
† GPDS - Dept. Enginyeria Electrònica. Universitat de València, Spain.
gustavo.camps@uv.es, http://www.uv.es/∼gcamps
‡ LEO - Dept. Termodinàmica. Universitat de València, Spain.

Abstract

The information contained in hyperspectral images allows the characterization, identification, and classification of the land-covers with improved accuracy and robustness. However, several critical problems should be considered in classification of hyperspectral images, among which: (i) the high number of spectral channels, (ii) the spatial variability of the spectral signature, (iii) the high cost of true sample labeling, and (iv) the quality of data. Many statistical and neural methods have been applied succesfully to this problem but some shortcomings are noticeable which have been recently alleviated by the introduction of kernel methods.

The chapter systematically discusses the specific problems and demands of this field, and reviews the most relevant works developed in our research group in hyperspectral image classification using kernel methods. We review the first attempts on neural and neurofuzzy approaches and how the introduction of kernel classifiers result in more accurate and robust outcomes. Also, we present a novel composite kernel-based approach which integrates the spatial and spectral domains simultaneously.

Keywords: Support vector machine, SVM, landcover classification, knowledge discovery, neural networks, composite kernel, hyperspectral, image classification, texture, contextual, spectral.

1 Introduction to Remote Sensing

Materials in a scene reflect, absorb, and emit electromagnetic radiation in a different way depending of their molecular composition and shape. Remote sensing exploits this physical fact and deals with the acquisition of information about a scene (or specific object) at a short, medium or long distance. The radiation acquired by an

* This research has been partially supported by the CICYT under Project DATASAT and by the "Grups Emergents" programme of Generalitat Valenciana under project HYPERCLASS/GV05/011.

(airborne or satellite) sensor is measured at different wavelengths and the resulting spectral signature (or *spectrum*) is used to identify a given material. The field of *spectroscopy* is concerned with the measurement, analysis, and interpretation of such spectra [1, 2].

Hyperspectral sensors are a class of imaging spectroscopy sensors acquiring hundreds of contiguous narrow bands or channels. Hyperspectral sensors sample the reflective portion of the electromagnetic spectrum ranging from the visible region (0.4-0.7μm) through the near-infrared to the near-infrared(about 2.4 μm) in hundreds of N narrow contiguous bands about 10 nm wide or less[1]. Hyperspectral sensors represent an evolution in technology from earlier multispectral sensors, which typically collect spectral information in only a few discrete, non-contiguous wide bands.

The high spectral resolution characteristic of hyperspectral sensors preserves important aspects of the spectrum (e.g., shape of narrow absorption bands), and makes differentiation of different materials on the ground possible. The spatially and spectrally sampled information can be described as a data cube (colloquially referred to as "the hypercube"), which includes two spatial coordinates and the spectral one (or wavelength). As a consequence, each image pixel is defined in a high dimensional space where each dimension corresponds to a given wavelength interval in the spectrum, $\mathbf{x}_i \in \mathbb{R}^N$, where N is the number of spectral channels or bands.

Remote sensing images acquired by previous generation multispectral sensors (such as the widely used Landsat Thematic Mapper sensor), have shown their usefulness in numerous Earth Observation (EO) applications. In general, the relatively small number of acquisition channels that characterizes multispectral sensors may be sufficient to discriminate among different land-cover classes (e.g., forestry, water, crops, urban areas, etc.). However, their discrimination capability is very limited when different types (or conditions) of the same species (e.g., different types of forest) are to be recognized. Hyperspectral sensors can be used to deal with this problem and represents a further step ahead in achieving the main general goals of remote sensing, which are:

1. *"Monitoring and modeling the processes on the Earth surface and their interaction with the atmosphere."*

2. *"Obtaining quantitative measurements and estimations of geo/bio/physical variables."*

3. *"Identifying materials on the land cover analyzing the acquired spectral signatures by satellite/airborne sensors."*

[1] Other types of hyperspectral sensors exploit the emissive properties of objects by collecting data in the mid-wave and long-wave infrared (MWIR and LWIR) regions of the spectrum.

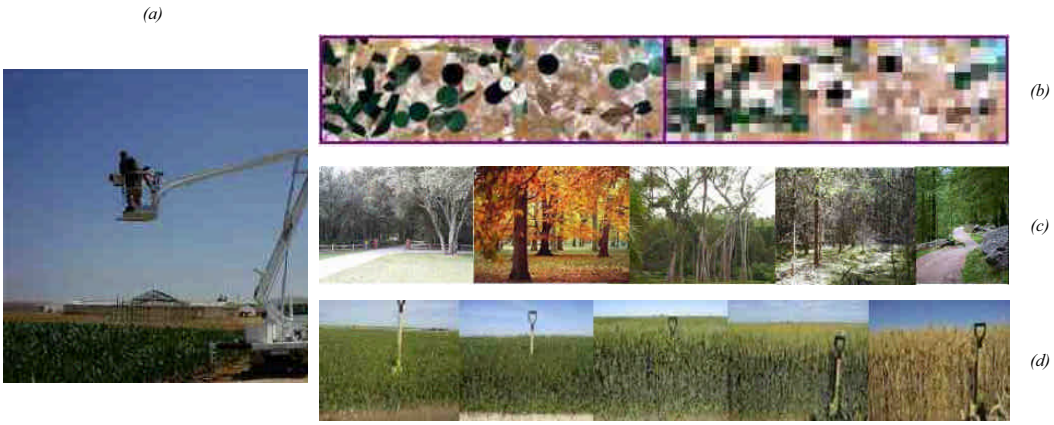


Figure 1: Illustrative examples of encountered problems in hyperspectral image classification: (a) a terrestrial campaign is necessary to accurately obtain a labeled training set, (b) different sensors provide different spectral and spatial resolutions from the same scene, (c) defining a class is sometimes difficult ('what is a forest?'), and (d) images from the same scene acquired at different time instants contain different spectral and spatial characteristics.

To attain such objectives, the remote sensing community has evolved to a multidisciplinary field of science that embraces physics, chemistry, biology, signal theory, computer science, electronics, and communications. From a *machine learning* and *signal/image processing* point of view, all these problems and applications are tackled under specific formalisms (classification, regression, modeling, image coding, spectral unmixing, etc) and among all of them, classification of hyperspectral images has become an important field of remote sensing.

2 Hyperspectral Image Classification. From neural to kernel methods

The information contained in hyperspectral images allows more accurate and robust characterization, identification, and classification of the land-covers [3]. Nevertheless, unlike multispectral data, hyperspectral images can not be analyzed by manual photo-interpretation or visual inspection, as the hundreds of available spectral channels (images) do not make it possible to accomplish this task. Consequently, many researchers have turned to techniques for addressing hyper-dimensional classification problems from the fields of statistics and machine learning in order to automatically generate reliable supervised and unsupervised classifiers. Unsupervised methods are

not sensitive to the ratio between number of labeled samples and number of features, since they work on the whole image, but the correspondence between clusters and desired classes is not ensured. Consequently, supervised methods are preferable when the desired input-output mapping is well-defined and a training set of true labels is available. However, several critical problems arise when dealing with the supervised classification of hyperspectral images:

1. The high number of spectral channels in hyperspectral images and the relatively low number of available labeled samples (due to the high cost of groud-truth collection process) poses the problem of *curse of dimensionality* or Hughes phenomenon [4] (see Fig. 1(a)).
2. The spatial variability of the spectral signature of each land-cover class (which is not stationary in the spatial domain) results in a critical variability of the values of feature-vector components of each class (see Figure 1(b)).
3. Uncertainty and variability on class definition (see Fig. 1(c)).
4. Temporal evolution of the Earth's cover (see Fig. 1(d)).
5. Illumination and athmospherical conditions, along with angular effects also increase the level of difficulty for classification.
6. The presence of different noise sources and uncertainties in the acquired image, e.g. in the measurement instrumental, observational noise, and uncertainty in the acquisition time.

In this context, *robust* and accurate classifiers are needed. In the remote sensing literature, many supervised methods have been developed to tackle the multispectral image classification problem. A successful approach to multispectral image classification is based on the use of artificial neural networks [5–8]. However, these approaches are not effective when dealing with a high number of spectral bands (Hughes phenomenon [4]), or when working with low number of training samples (ill-posed problems).

Much work has been carried out in the literature to overcome the aforementioned problem. Four main approaches can be identified: 1) regularization of the sample covariance matrix in statistical classifiers; 2) adaptive statistics estimation by the exploitation of the classified (semi-labeled) samples; 3) pre-processing techniques based on feature selection/extraction, aimed at reducing/transforming the original feature space into another space of a lower dimensionality; and 4) analysis of the behaviour of the spectral signatures to model the classes.

An elegant alternative approach to the analysis of hyperspectral data consists of kernel methods [9–11]. Many works have been presented in the last decade developing hyperspectral kernel classifiers. Support Vector Machines (SVMs) were

first applied to hyperspectral image classification in [12], and their capabilities were further analyzed in [13–18] in terms of stability, robustness to noise, and accuracy. Some other kernel methods have been recently presented to improve classification, such as Kernel PCA [14], the kernel Fisher discriminant (KFD) analysis [19], the regularized AdaBoosting [20], or Support Vector Clustering (SVC) [21, 22]. Lately, some kernel formulations have appeared in the context of target and anomaly detection [23,24], which basically consists of using the spectral information of different materials to discriminate between the target and background signatures. Finally, in [25], an extensive comparison of kernel-based classifiers (RBF neural networks, SVM, KFD, and regularized AdaBoosting) was conducted by taking into account the peculiarities of hyperspectral images, i.e. assessment was conducted in terms of the accuracy of methods when working in noisy environments, high input dimension, and limited training sets.

3 Classification of HyMap hyperspectral images with neural networks and SVMs

3.1 Neural and neurofuzzy networks

The traditional model of a feedforward multilayer neural network, commonly known as multilayer perceptron (MLP), is composed of a fully-connected layered arrangement of artificial neurons in which each neuron of a given layer feeds all the neurons of the next layer [26] (Fig. 2(a)). An MLP for multiclassification requires an output node for each class if no output coding is performed. Training of the network can be accomplished using the *backpropagation* learning algorithm [27].

In a Radial Basis Functions (RBF) neural network, notationally, the sigmoid-shape activation function of an MLP is substituted by a Gaussian function (Fig. 2(b)). The learning rule to update weight and variance vectors can be derived by using the *delta rule*. Gaussian-like RBFs are local, i.e. give a significant response only in a neighbourhood near the centre. These features induce good mappings but, in turn, may produce overfitting and yield poor results with uncertain inputs (noisy environments), and thus regularization becomes necessary [25].

A very promising paradigm in machine learning is constituted by the neurofuzzy approach in which, fuzzy logic and neural networks are combined. The Co-Active Neuro-Fuzzy Inference Systems (CANFIS) model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions (Fig. 2(c)). Fuzzy inference systems are also valuable as they combine the explanatory nature of rules (membership functions, MF) with the power of neural networks. These kinds of fuzzy networks solve problems more efficiently than common feedforward neural networks when the underlying function to model

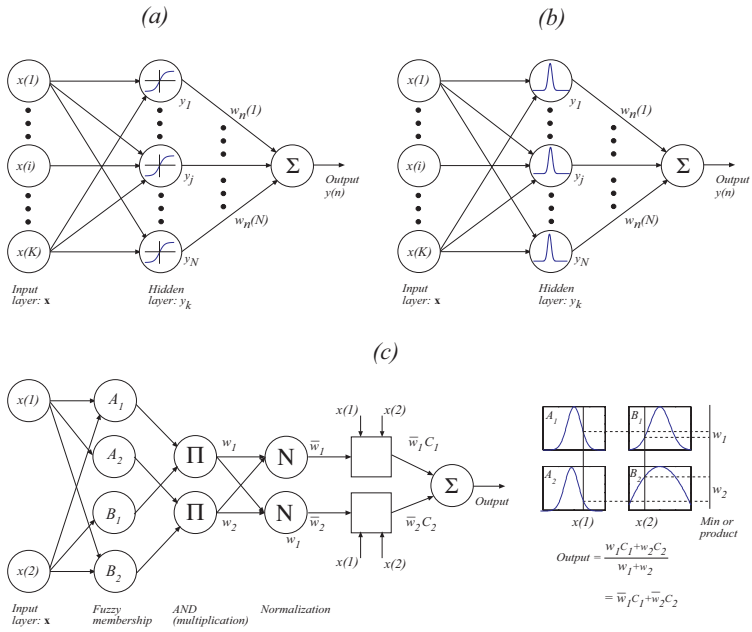


Figure 2: Schematic of the neural networks used in this work. (a) In an MLP, each neuron passes the weighted sum of its inputs through a sigmoid-shape function (e.g. hyperbolic tangent). The output of a neuron in a given layer acts as an input to neurons in the next layer. In the network illustration, each line represents a synaptic connection. (b) In an RBF neural network, the sigmoidal activation function of an MLP is replaced by a Gaussian function with adjustable widths and centers. (c) A two-input, one-output CANFIS network and an illustration of output calculation.

is highly variable or locally extreme since, in those cases, MLP or RBF networks attempt to discover a global optimization. The fundamental component of CANFIS is a fuzzy axon which applies membership functions to the inputs. Basically, two membership function types can be used (Gaussian or generalized bell). Fuzzy axons are valuable because their MF can be modified through backpropagation during network training to expedite the convergence. A second advantage is that fuzzy synapses aid in characterizing inputs that are not easily discretized. The second major component of CANFIS is a modular network that applies functional rules to the inputs. Two fuzzy structures are mainly used; the Tsukamoto model and the Sugeno (TSK) model. Finally, a combiner is used to apply the MF outputs to the modular network outputs. The combined outputs are then channeled through a final output layer and the error is backpropagated to both the MF and the modular network. Full details of this network can be found in [28].

3.2 Support Vector Machines

Neural networks and other gradient-descent based methods are trained in order to minimize the so-called *empirical risk*, i.e. the error in the training data set and, therefore, follow the Empirical Risk Minimization (ERM) principle. However, to attain significant results in the validation set ("out-of-sample" dataset), stopping-criteria or pruning techniques must be used. On the other hand, SVMs have been recently proposed as an efficient method for pattern classification and nonlinear regression. Their appeal lies in their strong connection to the underlying statistical learning theory where an SVM is an approximate implementation of the method of structural risk minimization (SRM) [9]. This principle states that a better solution (in terms of generalization capabilities) can be found by minimizing an upper bound of the generalization error. SVMs have many attractive features. For instance, the solution of the quadratic programming (QP) problem is globally optimized while, with neural networks, the gradient based training algorithms only guarantee finding a local minima. In addition, SVM, can handle large input spaces, which is especially convenient when working with hyperspectral data, can effectively avoid overfitting by controlling the margin, and can automatically identify a small subset made up of informative points, namely *support vectors* (SV).

The basic formulation for binary classification using SVMs is as follows. Given a labeled training data set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, and a nonlinear mapping $\boldsymbol{\phi}(\cdot)$, usually to a higher (possibly infinite) dimensional (Hilbert) space, $\boldsymbol{\phi} : \mathbb{R}^N \longrightarrow \mathcal{H}$, the SVM method solves:

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \right\} \tag{1}$$

constrained to:

$$y_i(\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{3}$$

where $\mathbf{w}$ and b define a linear classifier in the feature space. The non-linear mapping function $\boldsymbol{\phi}$ is performed in accordance with Cover's theorem [29], which guarantees that the transformed samples are more likely to be linearly separable in the resulting feature space. The regularization parameter C controls the generalization capabilities of the classifier and it must be selected by the user, and ξ_i are positive slack variables enabling to deal with permitted errors.

Due to the high dimensionality of vector variable $\mathbf{w}$, primal function (1) is usually solved through its Lagrangian dual problem, which consists of solving

$$\max_{\alpha_i} \left\{ \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle \right\} \tag{4}$$

constrained to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$, $i = 1, \ldots, n$, where auxiliary variables α_i are Lagrange multipliers corresponding to constraints in (2). It is worth noting that all $\boldsymbol{\phi}$ mappings used in the SVM learning occur in the form of inner products. This allows us to define a kernel function K:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle, \tag{5}$$

and then a non-linear SVM can be constructed using only the kernel function, without having to consider the mapping $\boldsymbol{\phi}$ explicitly. Then, by introducing (5) into (4), the dual problem is obtained. After solving this dual problem, $\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)$, and the decision function implemented by the classifier for any test vector $\mathbf{x}$ is given by $f(\mathbf{x}) = sgn\left(\sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$, where b can be easily computed from the α_i that are neither 0 nor C, as explained in [11].

3.3 Material and experimental setup

We used six hyperspectral images acquired with the 128-bands HyMap airborne spectrometer during the DAISEX-99 campaign (http://io.uv.es/projects/daisex/). More information about the data collection, Hymap calibration and atmospheric correction can be retrieved from [30]. Six different classes were considered in the area (corn, sugar beet, barley, wheat, alfalfa, and soil), which were labelled from ♯1 to ♯6, respectively. In this sense, the task is referred to as a multiclassification pattern recognition problem. Two data sets (training and validation sets) were built (150 samples/class each) and models were selected using the cross-validation method. Finally, a test set consisting of the true map on the scene over complete images was used as the final performance indicator. In each one of the six images (700×670 pixels), the total number of test samples is 327,336 (corn 31,269; sugar beet 11,322; barley 124,768; wheat 53,400; alfalfa 24,726; and bare soil 81,851) and the rest is considered unknown.

Once the desired input-output mapping for training and validation are defined, usually a feature selection stage is used to reduce dimension of the input space. This can make the training process feasible and improve results by removing noisy irrelevant bands. However, design and application of dimension-reduction techniques is time-consuming and scenario-dependent, which are evident problems to circumvent. In fact, we are not only interested in the classification accuracy provided by each method but also in their suitability to real-time working conditions whenever a feature selection stage is not possible. This scenario is simulated by considering models with and without a feature selection stage. The proposed learning scheme is shown in Fig. 3. In particular, previous work [30] in feature selection yielded three subsets of representative features (6, 3 and 2 bands), which induce three different pattern recognition problems, respectively.

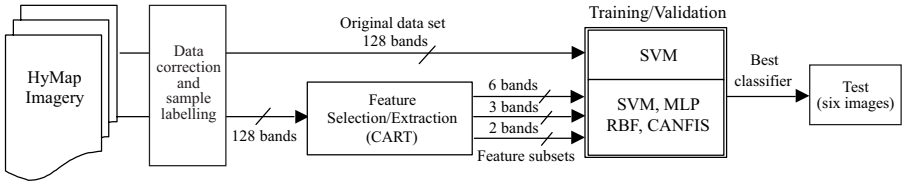


Figure 3: Diagram of the hyperspectral data classification process. A training data set is extracted from the the six collected images and then a CART-based feature selection stage yields three representative subsets (consisting of 6, 3 and 2 bands, respectively) [30], which constitute three different pattern recognition problems, respectively. An additional scenario considering the whole training data set (128 bands) incorporates. Four classifiers are thus implemented and tested in the six whole images.

3.4 Model development

As regards the MLP and RBF models, we varied the number of hidden neurons (< 100 to avoid overfitting), the weight initialization range and the learning rate (between 0.01 and 3) in order to determine the best topology. A great amount of CANFIS models were developed by varying the number (2-8) and structure (Bell and Gaussian) of the MF and the fuzzy model (TSK and Tsukamoto), along with the number of hidden layers (2-5) and step size (0.001-0.1). The momentum term remained constant and equal to zero.

In the case of SVMs, nonlinear classifiers were obtained by taking the dot product in kernel-generated spaces. The following kernels have been used in this work: (1) Linear: $K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$, (2) Polynomial: $K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$, and (3) Gaussian (RBF): $K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$. Note that one or more free parameters must be previously settled in the nonlinear kernels (polynomial degree d, Gaussian width γ) together with the *penalization* parameter C. In all cases, we considered equiprobable classes for training and validation and thus no individual penalization parameter was used [31]. However, the test set contains highly unbalanced classes and thus, the latter practice could improve results if the training process were intentionally driven by priors. However, this would not be a fair assumption for our purposes, i.e. achieving an automatic scenario-independent classifier.

The selection of the best subset of free parameters is usually done by cross-validation methods but this can lead to poor generalization capabilities and lack of representation. We alleviated this problem by using the 8-fold cross-validation method[2] with the training data set.

[2]The 8-fold cross–validation uses 7/8 of the data for training and 1/8 for validation purposes. This procedure is repeated eight times with different validation sets.

Many discriminative methods, including neural networks and SVMs, are often more accurate and efficient when dealing with only two classes. For large numbers of classes, higher-level multiclass methods utilize these two-class classification methods as the basic building blocks, namely "one-against-the-rest" procedures. However, such approaches lead to suboptimal solutions when dealing with multiclass problems and the well-known problem of the "false positives". Therefore, we have used a multiclassification scheme for all the methods.

All neural models were developed in MATLAB® environment (Mathworks, Inc). In the case of SVM, we used the libSVM implementation, which is freely available from http://www.csie.ntu.edu.tw/∼cjlin/.

3.5 Model comparison

Table 1 shows the average recognition rate (ARR[%]) of the six images in training, validation, and test sets. The ARR% is calculated as the rate of correctly classified samples over the total number of samples averaged over the six available images. Section 3.3 contains details on the training, validation and test sets.

Some conclusions can be drawn from Table 1. SVMs perform better than neural networks in all scenarios. Moreover, when a feature selection stage is not possible, and thus 128 bands should be used, the computational burden involved in the training process of neural networks make these methods unfeasible. In contrast, SVMs are not drastically affected by input dimension and presence of noisy bands. This has sometimes led to the idea that a feature selection is not necessary when working with SVMs, which is not completely true, as shown in [11,32]. In noisy applications, a feature selection is not only recommendable but mandatory, since it could remove undesired features and better results could thus be obtained. In our case study, no numerical (ARR<3%) or statistical (κ scores in the range [0.6,0.8]) differences are found between SVMs with and without a step for dimensionality reduction prior to classification. This indicates that noisy bands have been successfully identified and their contribution to the final decision attenuated without decreasing the recognition rate. Therefore, two preliminary conclusions can be extracted:

1. SVMs have proven to be efficient models that inherently detect noisy features.
2. A feature selection step slightly improves results.

This induces a clear trade-off: we could obtain good results by using an SVM without a preliminary feature selection stage or, we could (slightly) improve results by including a dedicated feature selection step, which is time-consuming and requires more effort. Depending on the application requirements, the user could choose between these two options.

Table 1: Average recognition rates (ARR [%]) of the six images in training, validation, and test sets for different models. The four subsets (128, 6, 3, 2 bands) are evaluated, all of them containing 150 samples per class. The column "Features" gives some information about the final models. For the case of SVMs, we indicate in brackets the penalization parameter, the kernel used and its optimal parameters (polynomial order d or Gaussian width γ), and the rate of support vectors, respectively. Bold face font is used to indicate the best kernel in each subset. For the case of neural networks, we indicate the number of input×hidden×output nodes.

METHOD	FEATS.	TRAIN.	VALID.	TEST
SVM128	Linear	99.89	98.78	95.45
SVM128	**Polynomial**	100	98.78	**95.53**
	(5.59, 4, 12.11%)			
SVM128	RBF	100	97.78	94.13
SVM6	Linear	99.89	99.33	94.44
SVM6	**Polynomial**	99.79	99.44	**96.44**
	(20.57, 4, 8.67%)			
SVM6	RBF	100	98.78	94.87
SVM3	Linear	89.00	87.22	81.31
SVM3	Polynomial	88.89	87.44	82.03
SVM3	**RBF**	91.22	91.00	**85.16**
	(35.94, 10^{-5}, 12.88%)			
SVM2	Linear	89.11	88.33	81.42
SVM2	Polynomial	89.11	88.33	82.55
SVM2	**RBF**	89.11	89.11	**82.68**
	(43.29, 10^{-2}, 16.88%)			
MLP128	-	-	-	-
MLP6	6×5×6	99.33	99.44	94.53
MLP3	3×25×6	90.22	87.67	82.97
MLP2	2×27×6	88.00	85.67	81.95
RBF128	-	-	-	-
RBF6	6×16×6	98.88	98.80	94.10
RBF3	3×31×6	88.20	87.00	81.44
RBF2	2×18×6	87.33	85.25	81.62
CANFIS128	-	-	-	-
CANFIS6	6×2×7×6	98.68	96.66	94.22
CANFIS3	3×3×12×6	89.20	88.77	81.64
CANFIS2	2×8×15×6	86.33	86.00	81.82

In the same table, we also observe that, as the dimension of the input space is lower, neural networks degrade more rapidly than SVMs do. In that sense, the

Figure 4: (a) RGB composite of the red, green and blue channels from 128-bands HyMAP image taken in June, 1999 of Barrax (Spain). (b) Map of the whole image classified with the labels of the classes of interest.

complexity[3] of all models increases as the input dimension decreases. In fact, RBF kernels and more than 15% of SVs are strictly necessary to attain significant results with less than six bands. Despite the fact that the polynomial kernel has been claimed to be specially well-suited for hyperspectral data classification [35], it has yielded results similar to the ones for the linear kernel in our case (see the next section for details).

Figure 4 shows the original and the classified samples for one of the collected images. Corn classification seems to be the most troublesome. The reason for that is the presence of a whole field of two-leaf corn in the early stage of maturity, where soil was predominant and was not accounted for the reference labelled image. The confusion matrix supports this conclusion as most of the errors are committed with the bare soil class.

4 Classification of AVIRIS hyperspectral images with composite kernels

The good classification performance demonstrated by SVMs (cf. Section 3) using the spectral signature as input features can be further increased by including contextual (or even textural) information in the classifier. This can be easily carried out by

[3]We evaluate the model's complexity in terms of the kernel used and the number of SVs in the SVM approach, and in terms of the number of hidden neurons in the neural networks. We have based this decision on the works [11, 33, 34], where an intuitive relation between neural networks and Support Vector Machines is sketched.

means of the composite kernels framework, in which one exploits the properties of Mercer's kernels.

4.1 Composite kernels formulation

For this purpose, a pixel entity $\mathbf{x}_i$ is redefined simultaneously both in the spectral domain using its spectral content, $\mathbf{x}_i^\omega \in \mathbb{R}^{N_\omega}$, and in the spatial domain by applying some feature extraction to its surrounding area, $\mathbf{x}_i^s \in \mathbb{R}^{N_s}$, which yields N_s spatial (contextual) features, e.g. the mean or standard deviation *per* spectral band. These separated entities lead to two different kernel matrices, which can be easily computed using any suitable kernel function that fulfills Mercer's conditions. At this point, one can sum spectral and textural dedicated kernel matrices (K_ω and K_s, respectively), and introduce the cross-information between textural and spectral features ($K_{\omega s}$ and $K_{s\omega}$) in the formulation. This simple methodology yields a full family of composite methods for hyperspectral image classification [36], which can be summarized as follows:

- **The stacked features approach.** Let us define the mapping $\boldsymbol{\phi}$ as a transformation of the concatenation $\mathbf{x}_i \equiv \{\mathbf{x}_i^s,\ \mathbf{x}_i^\omega\}$, then the corresponding 'stacked' kernel matrix is:

$$K_{\{s,\omega\}} \equiv K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle, \tag{6}$$

  which does not include explicit cross relations between $\mathbf{x}_i^s$ and $\mathbf{x}_j^\omega$.

- **The weighted summation kernel.** Let us define (a weighted) concatenation of nonlinear transformations of $\mathbf{x}_i^s$ and $\mathbf{x}_i^\omega$, which finally yield the following composite kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \mu K_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + (1-\mu) K_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega)$$

  where μ is a positive real-valued free parameter ($0 < \mu < 1$), which is tuned in the training process and constitutes a trade-off between the spatial and spectral information to classify a given pixel.

- **The cross-information kernel.** Finally, one can define a weighted sum of positive definite matrices, accounting for the textural, spectral, and cross-terms between textural and spectral counterparts:

$$K(\mathbf{x}_i, \mathbf{x}_j) = K_s(\mathbf{x}_i^s, \mathbf{x}_j^s) + K_\omega(\mathbf{x}_i^\omega, \mathbf{x}_j^\omega) + K_{s\omega}(\mathbf{x}_i^s, \mathbf{x}_j^\omega) + K_{\omega s}(\mathbf{x}_i^\omega, \mathbf{x}_j^s) \tag{7}$$

4.2 Data collection

Experiments were carried out using the familiar AVIRIS image taken over NW Indiana's Indian Pine test site in June 1992 [37]. Following [12], we first used a part of the 145×145 scene, called the *subset scene*, consisting of pixels [27-94]×[31-116] for a size of 68×86, which contains four labeled classes (the background pixels were not considered for classification purposes). Second, we used the *whole scene*, consisting of the full 145×145 pixels, which contains 16 classes, ranging in size from 20 pixels to 2468 pixels. We removed 20 noisy bands covering the region of water absorption, and finally worked with 200 spectral bands. In both datasets, we used 20% of the labeled samples for training and the rest for validation.

4.3 Model development

In all cases, we used the polynomial kernel ($d = \{1, \ldots, 10\}$) for the spectral features according to previous results [12,16], and used the RBF kernel ($\sigma = \{10^{-1}, \ldots, 10^{3}\}$) for the spatial features according to the locality assumption in the spatial domain. In the case of the weighted summation kernel, μ was varied in steps of 0.1 in the range [0,1]. For simplicity and for illustrative purposes, μ was the same for all labeled classes in our experiments. For the 'stacked' ($K_{\{s,\omega\}}$) and cross-information ($K_{s\omega}$, $K_{\omega s}$) approaches, we used the polynomial kernel. The penalization factor in the SVM was tuned in the range $C = \{10^{-1}, \ldots, 10^{7}\}$. A *one-against-one* multiclassification scheme was adopted in both cases.

The most simple but powerful spatial features $\mathbf{x}_i^s$ that can be extracted from a given region are based on moment criteria. In this chapter, we take into account the first two momenta to build the spatial kernels. Two situations were considered: (i) using the mean of the neighborhood pixels in a *window* ($dim(\mathbf{x}_i^s) = 200$) *per* spectral channel or (ii) using the mean and standard deviation of the neighborhood pixels in a *window* per spectral channel ($dim(\mathbf{x}_i^s) = 400$). Inclusion of higher order momenta or cumulants did not improve the results in our case study. The *window* size was varied between 3×3 and 9×9 pixels in the training set.

4.4 Model comparison

Table 2 shows the validation results of several classifiers for both images (averaged over 10 random realizations that were obtained to avoid skewed conclusions). We include results from six kernel classifiers: spectral (K_ω), contextual (K_s), the stacked approach ($K_{\{s,\omega\}}$), and the three presented composite kernels. In addition, two standard methods are included for baseline comparison: bLOOC + DAFE + ECHO, which uses contextual and spectral information to classify homogeneous objects, and the Euclidean classifier [38], which only uses the spectral information. All models

Table 2: Overall accuracy, OA[%], and kappa statistic, κ, on the validation sets of the subset and whole scenes for different spatial and spectral classifiers. The best scores for each class are highlighted in bold face font. The OA[%] that are statistically different (at 95% confidence level, as tested through paired Wilcoxon rank sum test) from the best model are underlined.

	SUBSET SCENE		**WHOLE SCENE**	
	OA[%]	κ	**OA**[%]	κ
Spectral classifiers[†]				
Euclidean [38]	<u>67.43</u>	—	<u>48.23</u>	—
bLOOC+DAFE+ECHO [38]	<u>93.50</u>	—	<u>82.91</u>	—
K_ω [12]	95.90	—	<u>87.30</u>	—
K_ω developed in this chapter	95.10	0.94	<u>88.55</u>	0.87
Spatial-spectral classifiers				
Mean				
K_s	93.44	0.92	<u>84.55</u>	0.82
$K_{\{s,\omega\}}$	96.84	0.97	94.21	0.93
$K_s + K_\omega$	97.12	0.97	92.61	0.91
$\mu K_s + (1-\mu)K_\omega$	97.43	0.97	**95.97**	**0.94**
$K_s + K_\omega + K_{s\omega} + K_{\omega s}$	**97.44**	**0.97**	94.80	0.94
Mean and standard deviation [‡]				
K_s	94.86	0.94	<u>88.00</u>	0.86
$K_{\{s,\omega\}}$	98.23	0.97	94.21	0.93
$K_s + K_\omega$	98.26	0.98	95.45	0.95
$\mu K_s + (1-\mu)K_\omega$	**98.86**	**0.98**	**96.53**	**0.96**

[†] One difference with the data and results reported in [38] is that they studied the scene using 17 classes (Soybeans-notill was split into two classes) whereas we used 16 classes. Also note that the use of the LOOC algorithm instead of the bLOOC algorithm could improve performance, as proposed in [39,40]. Differences between the obtained accuracies reported in [12] and the presented here could be due to the random sample selection, however they are not statistically significant. [‡] Note that by using mean and standard deviation features, $N_\omega \neq N_s$ and thus no cross kernels ($K_{s\omega}$ or $K_{\omega s}$) can be constructed.

are compared numerically (overall accuracy, OA[%]) and statistically (kappa test and Wilcoxon rank sum test).

Several conclusions can be obtained from Table 2. First, all kernel-based methods produce better (and statistically significant) classification results than previous methods (simple Euclidean and LOOC-based method), as previously illustrated in [12]. It is also worth noting that the contextual kernel classifier K_s alone produces good results in both images, mainly due to the presence of large homogeneous classes and the high spatial resolution of the sensor. Note that the extracted textural features $\mathbf{x}_i^s$ contain spectral information to some extent as we computed them

per spectral channel, thus they can be regarded as contextual or local spectral features. However, the accuracy is inferior to the best spectral kernel classifiers (both K_ω implemented here and in [12]), which demonstrates the relevance of the spectral information for hyperspectral image classification. Furthermore, it is worth mentioning that all composite kernel classifiers improved the results obtained by the usual spectral kernel, which confirms the validity of the presented framework. This improvement was higher in the most difficult case of the whole scene (11% increase vs. 4% in the subset image) since the spatial variability of the spectral signature was reduced, and classifiers take advantage of the spatial correlation to enhance their accuracy by correctly identifying neighboring classes.

The good numerical and statistical results obtained can be assessed by showing the best classified images in Fig. 5 (whole scene). It is worth noting that narrow inter-class boundaries are smoothed and better discerned with the inclusion of composite kernels. Finally, two relevant issues should be highlighted from the obtained results: (i) optimal μ and window size seem to act as efficient alternative trade-off parameters to account for the textural information ($\mu = 0.2$ and 7×7 for the subset image, $\mu = 0.4$ and 5×5 for the whole image), and (ii) results have been significantly improved without considering any feature selection step previous to model development. These findings should be further explored in more applications and scenarios. In conclusion, composite kernels offer excellent performance for the classification of hyperspectral images by simultaneously exploiting both the spatial and spectral information.

5 Discussion and Conclusions

In this chapter, we have revised our experience in developing neural and kernel methods for hyperspectral image classification. Many experiments have been presented, and a novel formulation that efficiently integrates the spatial and spectral information has been considered to improve performance. The family of composite kernels offers an elegant kernel formulation to integrate spatial and spectral information, and opens a wide field for further developments.

In conclusion, we can state that, in the standard situation and in our case studies, the use of SVMs is more beneficial than neural networks, mainly because they work efficiently with high input dimension samples, they ensure sparsity (over the samples), and they have very few free parameters to tune. However, it is worth noting that in order to attain significant results, the standard algorithm of SVMs must be tailored to exploit the special characteristics of hyperspectral images, as presented in the composite framework.

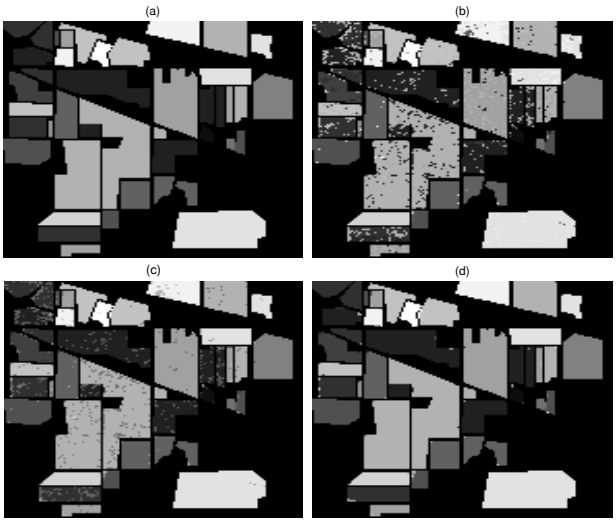


Figure 5: Classification results in the *whole image*. (a) Labeled scene and classification maps using the (b) contextual kernel, K_s (window size: 5×5), (c) spectral kernel, K_ω, and (d) weighted summation kernel ($\mu K_s + (1-\mu)K_\omega$, μ=0.4, window size: 5×5).

References

[1] J. A. Richards and Xiuping Jia. *Remote Sensing Digital Image Analysis. An Introduction*. Springer-Verlag, Berlin, Heidenberg, 3rd edition, 1999.

[2] G. Shaw and D. Manolakis. Signal processing for hyperspectral image exploitation. *IEEE Signal Processing Magazine*, 50:12–16, Jan 2002.

[3] P.H. Swain. *Remote Sensing: The Quantitative Approach*, chapter Fundamentals of pattern recognition in remote sensing, pages 136–188. McGraw-Hill, New York, NY, 1978.

[4] G. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.

[5] H. Bischof and A. Leona. Finding optimal neural networks for land use classification. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):337–341, 1998.

[6] H. Yang, F. van der Meer, W. Bakker, and Z. J. Tan. A back–propagation neural network for mineralogical mapping from AVIRIS data. *International Journal of Remote Sensing*, 20(1):97–110, 1999.

[7] G. Giacinto, F. Roli, and L. Bruzzone. Combination of neural and statistical algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters*, 21(5):399–405, 2000.

[8] L. Bruzzone and R. Cossu. A multiple-cascade-classifier system for a robust and partially unsupervised updating of land-cover maps. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9):1984–1996, 2002.

[9] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000. http://www.support-vector.net.

[11] B. Schölkopf and A. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press Series, 2002.

[12] J. A. Gualtieri, S. R. Chettri, R. F. Cromp, and L. F. Johnson. Support vector machine classifiers as applied to AVIRIS data. In *Proceedings of The 1999 Airborne Geoscience Workshop*, February 1999.

[13] C. Huang, L. S. Davis, and J. R. G. Townshend. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4):725–749, 2002.

[14] G. Camps-Valls, L. Gómez-Chova, J. Calpe, E. Soria, J. D. Martín, and J. Moreno. Kernel methods for HyMap imagery knowledge discovery. In *SPIE International Symposium Remote Sensing*, Barcelona, Spain, Set 2003.

[15] L. Bruzzone and F. Melgani. Classification of hyperspectral images with support vector machines: multiclass strategies. In *SPIE International Symposium Remote Sensing IX*, pages 408–419, Barcelona, Spain, Set 2003. SPIE.

[16] G. Camps-Valls, L. Gómez-Chova, J. Calpe, E. Soria, J. D. Martín, L. Alonso, and J. Moreno. Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1530–1542, July 2004.

[17] F. Melgani and L. Bruzzone. Classification of hyperspectral remote-sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778– 1790, Aug 2004.

[18] G. M. Foody and J. Mathur. A relative evaluation of multiclass image classification by support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–9, July 2004.

[19] M. Dundar and A. Langrebe. A cost-effective semisupervised classifier approach with kernels. *IEEE Transactions on Geoscience and Remote Sensing*, 42(1):264–270, January 2004.

[20] G. Camps-Valls, A. Serrano-López, L. Gómez-Chova, J. D. Martín, J. Calpe, and J. Moreno. Regularized RBF networks for hyperspectral data classification. In *International Conference on Image Recognition, ICIAR 2004*, Porto, Portugal, Oct 2004. Lecture Notes in Computer Science. Springer-Verlag.

[21] A. N. Srivastava and J. Stroeve. Onboard detection of snow, ice, clouds and other geophysical processes using kernel methods. In *Proceedings of the ICML 2003 Workshop on Machine Learning Technologies for Autonomous Space Sciences*, Washington, DC USA, August 2003.

[22] X. Song, G. Cherian, and G. Fan. A ν-insensitive SVM approach for compliance monitoring of the conservation reserve program. *IEEE Geoscience and Remote Sensing Letters*, 2(2):99–103, April 2005.

[23] H. Kwon and N.M. Nasrabadi. Hyperspectral target detection using kernel matched subspace detector. In *International Conference on Image Processing, ICIP'04*, pages 3327–3330, October 2004.

[24] H. Kwon and N.M. Nasrabadi. Hyperspectral anomaly detection using kernel RX-algorithm. In *International Conference on Image Processing, ICIP'04*, pages 3331–3334, October 2004.

[25] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351– 1362, June 2005.

[26] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Englewood Cliffs, NJ, 1999.

[27] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. Cambridge, MA: MIT Press, 1986.

[28] Jang Jyh-Shing Roger, Sun Chuen-Tsai, and Mizutani Eiji. *Neuro–Fuzzy and Soft–Computing*. Prentice Hall, Englewood Cliffs, NJ, 1997.

[29] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334, June 1965.

[30] L. Gómez-Chova, J. Calpe, E. Soria, G. Camps-Valls, J. D. Martín, and J. Moreno. CART-based feature selection of hyperspectral images for crop cover classification. In *IEEE International Conference on Image Processing*, 2003. Submitted.

[31] Y. Lin, Y. Lee, and G. Wahba. Support Vector Machines for classification in nonstandard situations. Department of Statistics TR 1016, University of Wisconsin-Madison, 2000. http://www.kernel-machines.org/.

[32] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.

[33] B. Schölkopf, K.-K. Sung, C. J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. N. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Trans. on Signal Processing*, 45(11):2758–2765, 1997.

[34] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

[35] J. A. Gualtieri and R. F. Cromp. Support vector machines for hyperspectral remote sensing classification. In *Proceedings of the SPIE, 27th AIPR Workshop*, pages 221–232, February 1998.

[36] G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla. Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters, In press*, Nov 2005.

[37] D. Landgrebe. AVIRIS NW Indiana's Indian Pines 1992 data set, 1992. http://dynamo.ecn.purdue.edu/∼biehl/MultiSpec/documentation.html.

[38] S. Tadjudin and D. Landgrebe. *Classification of High Dimensional Data with Limited Training Samples*. PhD thesis, School of Electrical Engineering and Computer Science, Purdue University, May 1998. TR-ECE-98-9.

[39] Q. Jackson and D. A. Landgrebe. An adaptive method for combined covariance estimation and classification. *IEEE Transactions on Geoscience and Remote Sensing*, 40(5):1082–1087, May 2002.

[40] B-C. Kuo and D. A. Landgrebe. A covariance estimator for small sample size classification problems and its application to feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40(4):814–819, 2002.

ITI Image Recognition and Artificial Vision Group Activities *

J. Arlandis, J. Cano, J. García-Hernández,
R. Llobet, G. Mainar, R. Paredes,
A. Pérez, J. C. Pérez Cortés,
I. Salvador, A. Toselli, M. Villegas
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{arlandis,jcano,jgarcia,rllobet,gmainar,rparedes}@iti.upv.es
{jcperez,aperez,jpla,issalig,ahector,mvillegas}@iti.upv.es

Abstract

The Image Recognition and Artificial Vision group of the "Instituto Tecnologico de Informática" is a part of a larger group (Pattern Recognition and Human Language Technologies, PRHLT) in the same institution, focused on the field of image analysis and computer vision under the Pattern Recognition Paradigm.

The group has been especially targeted to computer vision applications, and their members have published a number of scientific papers and participated in a variety of projects along the last 15 years. Some examples of tasks dealt with in these projects are: industrial continuous material inspection, complex image analysis, optical/intelligent character recognition, colour recognition and other related areas like biometric identification (fingerprint, face and speaker recognition).

1 Introduction

A main R+D work line at the ITI is focused in Computer Vision. The *Image Recognition and Artificial Vision* group (RIVA) has a large experience in the field of image analysis. It is demonstrated by the publication of scientific papers and its participation in a variety of projects. The work developed by the group members is mainly centered in the fields of Pattern Recognition and Perception Technologies. The group has experience in tasks as continuous material inspection, complex image analysis, optical/intelligent character recognition, colour recognition and other

* Work funded by the Agencia Valenciana de Ciencia y Tecnología (AVCiT, ayuda para Grupos I+D+I, GRUPOS03/031) and by the Comisión Interministerial de Ciencia y Tecnología (CICYT, TIC 2003-08496-C04)

related areas as biometric identification (fingerprint, face, palmprint and speaker recognition).

2 Research Areas

In this section, the main research areas from the RIVA group are described in detail.

2.1 Document analysis

In many tasks a document digitalisation is needed. The problem to solve can differ significantly for different sources of documents. For instance, the documents can be handwritten or printed, the areas where the text is located can be previously known or, on the opposite case, they can arise at random points, etc.

Several members of the RIVA group have taken part in projects related with handwritten and printed optical character recognition [1, 2, 3, 4, 11]. As a results, a proprietary character recognition engine has been developed, as well as different tools designed "ad-hoc" to perform each one of the tasks that arose in each of the collaborations carried out with private companies skilled in this kind of tasks.

2.2 Medical image

One field in which digital image processing is providing invaluable help is medical image analysis. Due to the great responsibility of tasks in this field, current computer tools are designed to assist the professionals at their work and never substitute them [12].

For instance, the ITI has collaborated with local hospitals in the design of assistance tools to the diagnostic of prostate cancer with ultrasonographic images [14, 8]. Another similar tool is currently at experimentation stage and it's aimed to assist with the diagnostic of breast cancer from radiographic images [9, 10].

Finally, other group members are working in a collaboration project with another technological institute, skilled in biomechanics, to develop an assistance tool to the diagnostic of foot pathologies with the information extracted from pressure signal data [7].

2.3 Scene analysis

Different computer vision tasks, in which the target is to recognise an object or to identify a person, can be viewed as the result of a two-step image processing

techniques. While the first one is the responsible for the location of the objects of interest in a scene image (scene analysis) [13, 5], the second processing step should be able to recognise the objects previously selected (recognition). Thus, it has to be noted that scene analysis is a wide field that connects with a great variety of applications. Consequently, solutions devised to solve this problem follow very different approximations.

The RIVA group has experience in license plate and face detection in natural images, that is, designed to work in a wide range of acquisition conditions, including unrestricted scene environments, light, perspective and camera-to-object distance. This means that the complexity to locate an object in an image increases by variable illumination, perspective and background conditions.

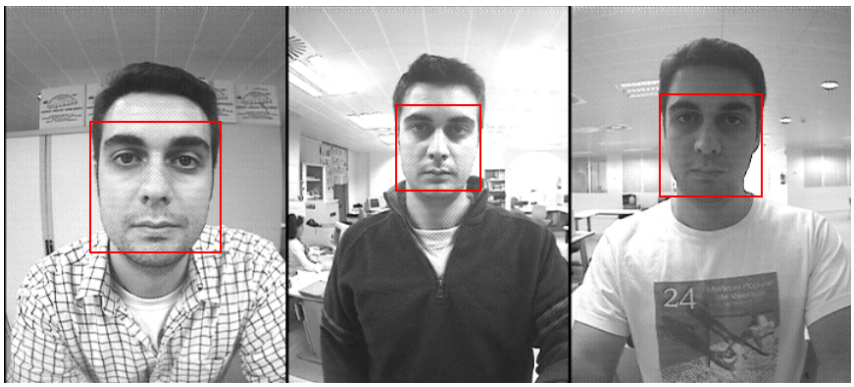


Figure 1: Scene analysis. Face segmentation

2.4 Industrial inspection

In this field, for example, there are many control processes without contact whose restrictions or features do not allow for the use of conventional tools available on the market. Among them: dimensional control process, measure, texture specification parametrisation, shape, colour, all kind of manufacturing defects, like foreign elements, dents, cracks, imperfections, etc

When one special feature of the process hinders the use of a commercial inspection product ("off the shelf"), often aimed at simple tasks, the application becomes

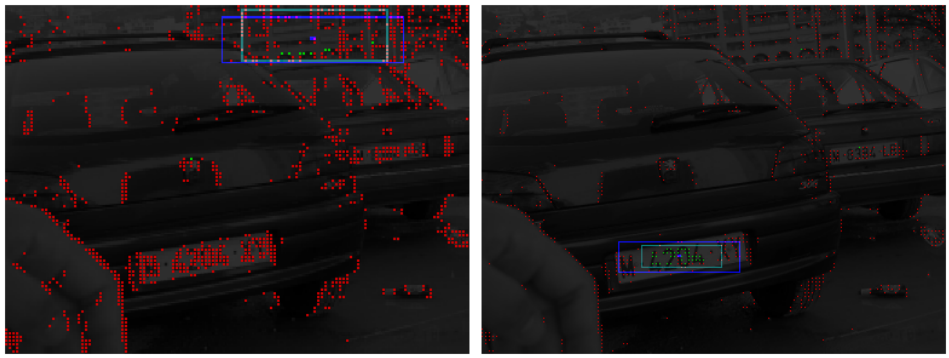


Figure 2: Scene analysis. Plate segmentation

a potential R+D project.

Several members of the RIVA group took part in the design, development and installation, in collaboration of company staff, of an inspection machine devoted to the detection of textile printing defects.

3 Projects

In this section, a set of projects in collaboration with corporations are described. In these projects the knowledge acquired at the pattern recognition field by the artificial vision group are applied to solve real problems.

3.1 Handwritten text recognition (Document analysis)

Continuous handwriting text recognition is yet a challenge. Although text is basically composed of individual characters, many approximations to optical character recognition do not achieve good results due to the extreme complexity of continuous handwriting segmentation [17].

Nevertheless, human beings are able to easily segmentate and recognise handwritten text. A way to achieve precision is to postpone recognition to a higher level. A sentence can be better understood when it has been completely read. This means a cooperative work among morphologic, lexical and syntactical levels that is

performed by continuous speech recognition techniques.

This methodology employs robust and validated algorithms. Moreover, a previous segmentation is not required as it is automatically achieved by decodification.

A number of members of the Artificial Vision Group have collaborated with private companies in the development of experimental systems able to recognise numeric quantities written on bank checks and forms with written polls without any language restrictions.

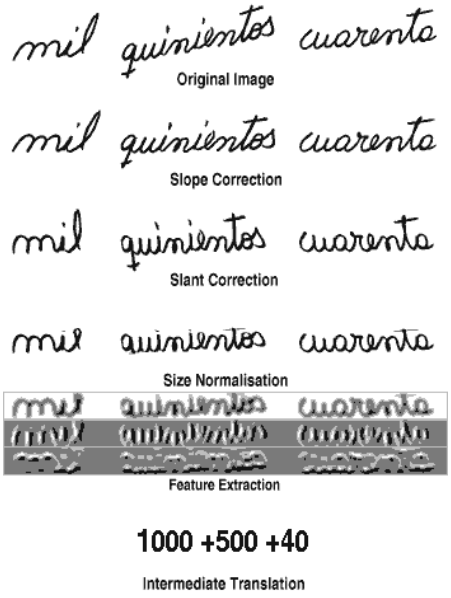


Figure 3: Document analysis. Handwritten text recognition.

3.2 Handwritten form recognition (Document analysis)

The system developed at the ITI takes advantage of OCR algorithms based on statistical classification methods in order to extract alphanumeric information from the form fields. The characters are automatically extracted from the handwritten form fields. The use of skilled models, automatically trained from samples, allows the system to work with any language and alphabet.

In main lines, the used preprocessing for digitalising a handwrite document can be divided in these three steps:

- Preprocess: the fields and cells get isolated by a segmentation process. This implies different image preprocess steps: noise removal, blank detection, minimum inclusion box definition and scale normalisation.
- Classification: Each isolated character is individually classified by the OCR engine.
- Parsing: Each recognised string in a field is submitted to a syntactic analysis process that rectifies, if it is needed, the original string to adjust it to a given language model [15]. Finally, a corrected string and a confidence value is provided by the system as the result of the whole recognition process.

The ITI has participated in several collaboration projects with private companies of the Valencian Community involved on automatic processing of thousand of text documents, as the elaboration of the census or the digitalisation of official documents (birth, marriage and decease bulletins).

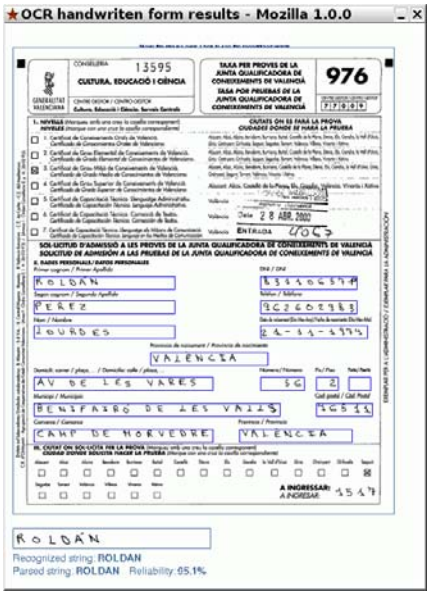


Figure 4: Document analysis. Handwritten form recognition

3.3 License plate recognition (Image analysis)

A license plate recognition engine designed to work with no restricted images (variable illumination, perspective and background) is available.

At the segmentation stage areas of a texture similar to a license plate are searched; this process produces a number of points classified as "plate" and others classified as "no plate". After that, a postprocess is applied to the points classified as plate, grouping them together in one or more clusters and the area with higher confidence to belong to a plate is returned as the plate segmentation hypothesis [5].

Finally, a multiple classification process is carried out over a set of pixels inside the plate hypothesis. This classification process provides with a character string that should be corrected by a known language model: the license plate format. Individual classification errors can be rectified applying a syntactic analyser. As a result, the recognition engine provides the plate identifier and a confidence level.

Several members of the Artificial Vision group have experience with private companies working in this kind of application. A common effort is being made among them to design and develop a whole license plate recognition system, aimed to be installed at the accessing points of a public parking.

3.4 Assistance to Prostate Cancer Detection (Medical Image)

The target of this project is to develop an automatic assistance system to the prostate cancer diagnosis from ultrasonographic images by means of image analysis and pattern recognition techniques. This tool can help the professional expert in the decision to realize a biopsy.

In order to discriminate between benign prostate diseases and malignant tumors, a diagnosis test known as TRUS (TransRectal UltraSonography imaging) can be used. A possible way to improve this TRUS-guided biopsying process is to use computer-aided analysis of the ultrasonographic image. The basic idea is to develop a computer-aided tool capable of highlighting the areas most likely to contain cancer cells. A training of the system is supervised by selecting the previous image to the biopsy (puncture) and labelling the biopsied area and the whole prostate. These labelled samples are finally used as the training or test data.

Texture classification (cancer/no-cancer) can be obtained by:

1. A confidence value of a fast neighbors search.
2. The probability of a hidden Markov model that models the two classes (cancer/no-cancer).

Candidate cancer areas are coloured in a way that the proposed puncture area can be easily seen.

Some of the members of the Artificial Vision group have large experience in this field thanks to the collaboration with the Urology Department of one of the main hospitals from Valencia.

3.5 Textile Quality Control (Industrial Inspection)

One of the first collaboration projects between ITI and a private company was targeted to design, develop and build a whole system to perform a quality control task in the textile printing process. The problem consisted on searching for printing defects. Finally an inspection tool was built, that allowed the human operators of the printing machine to register the first meters of the printed fabric and automatically check for repetitive defects on the rest of the printed fabric.

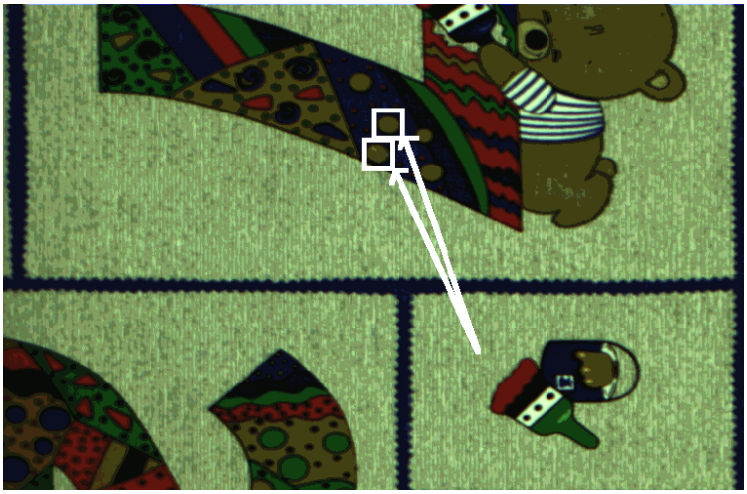


Figure 5: Industrial Inspection. Detection of defects on the printed fabric.

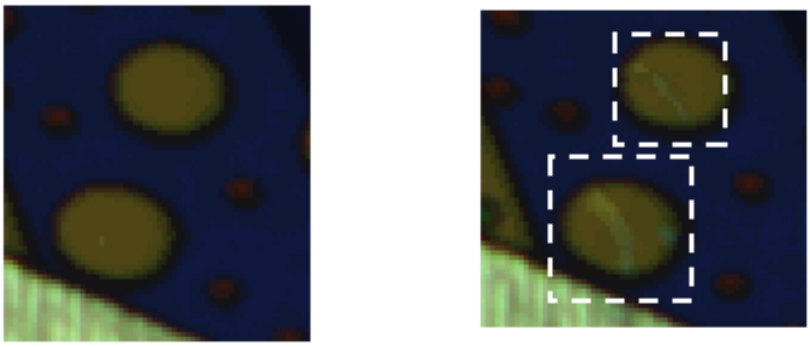


Figure 6: Industrial Inspection. Defect detail.

The textile printing process is a complex task. Among other things a precise synchronism of the printing cylinders, as well as perfectly homogeneous dye supply is required. Due to the previous highly demanding conditions, the appearance of repetitive defects in the printed web is sadly frequent.

The development of small adherences of threads can be sometimes confused with the texture of the printed pattern. A thread adherence blocks the dye printed onto the fabric, producing an area of a brighter colour. The width of the web can reach 3.6 meters, making necessary the use of 4 lineal cameras in order to achieve enough resolution to detect a defect as thin as a thread. Due to the mechanical stress incurred in the high speed of the printing process, the printed fabric experiments elastic distortions. This problem can be compensated by a local elastic registering technique [16]. Each pixel of the reference image is represented by a feature vector of high dimensionality that stores the colour features of the pixel neighborhood.

References

[1] Arlandis J., Pérez-Cortés J.C., Llobet R. *Handwritten Character Recognition Using Continuos Distance Transformation*, Proceedings of the 15th. International Conference on Pattern Recognition, 2000.

[2] Arlandis J., Pérez-Cortés J.C., *The Continuos Distance Transformation: A Generalization of the Distance Transformation for Continuos-valued Images*, Pattern

Recognition and Applications, 2000.

[3] Arlandis J., Pérez-Cortés J.C., *Fast Handwritten Recognition Using Continuous Distance Transformation, Progress in Pattern Recognition Speech and Image Analysis*, Lecture Notes in Computer Science (2905), 2003.

[4] Cano J., Pérez-Cortés J.C., Arlandis J., Llobet R., *Training Set Expansion in Handwritten Character Recognition*, International Workshop on Statistical Pattern Recognition, 2002.

[5] Cano J., Pérez-Cortés J.C., *Vehicle License Plate Segmentation In Natural Images*, Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), 2003.

[6] Cano J., Pérez-Cortés J.C., Salvador I., *Comparison Of Two Fast Nearest-Neighbour Search Methods in High-Dimensional Large-Sized Databases*, Workshop on Statistical Pattern Recognition, 2005.

[7] García-Hernández J., Paredes R., Garrido D., Soler C., *Foot pathologies classification pressure distribution over the foot plant*, In Procs. of the First International Workshop on Biosignal Processing and Classification (BPC 2005).

[8] Llobet R., Toselli A. H., Pérez-Cortés J. C., Juan A., *Computer-aided Prostate Cancer Detection in Ultrasonographic Images*, Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis, 2003.

[9] Llobet R., Toselli A. H., Pérez-Cortés J. C., *Breast Cancer Detection in Digitized Mammograms Using Non-Parametric Methods*, Proceedings of the 2nd International Conference on Advances in Biomedical Signal and Information Processing, 2004.

[10] Llobet R., Paredes R., Pérez-Cortés J.C., *Comparison of feature extraction methods for breast cancer detection*, 2nd Iberian Conference on Pattern Recognition and Image Analysis, 2005.

[11] Keysers D., Paredes R., Ney H., Vidal E., *Combination of Tangent Vectors and Local Representations for Handwritten Digit Recognition*, International Workshop on Statistical Pattern Recognition, 2002.

[12] Paredes R., Keysers D., Lehmann T., Wein B. B., Ney H., Vidal E., *Classification of Medical Images using Local Representations*, Bildverarbeitung für die Medizin, 2002.

[13] Deselaers T., Keysers D., Paredes R., Vidal E., Ney H., *Local Representations for Multi-Object Recognition*, Pattern Recognition, 25th DAGM Symposium, 2003.

[14] Pérez-Cortés J.C., Juan A., Vallada E., *Textural Analysis Of Prostate Cancer In Transrectal Ultrasound Images*, Proc. of Biosignal, 2002.

[15] Perez-Cortes J.C., Amengual J.C., Arlandis J., Llobet, R., *Stochastic Error Correcting Parsing for OCR Post-processing*, International Conference on Pattern Recognition, 2000.

[16] Perez-Cortes J.C., Paredes R., Valiente J.M., Arlandis J., Cano J., *An Elastic Registration Method for Quality Control of Textile Printing*, Pattern Recognition and Image Analysis, 1999.

[17] Toselli A., Juan A., Vidal E., *Spontaneous Handwriting Recognition and Classification*, Proceedings of the 17th International Conference on Pattern Recognition, 2004.

OCR Research in PRHLT Group *

J. García-Hernández A.H. Toselli J. Arlandis R. Paredes
R. Llobet A. Juan J.C. Pérez Cortés J. Cano A. Pérez
E. Vidal F. Casacuberta
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
{jgarcia,ahector,arlandis,rparedes,rllobet}@iti.upv.es
{ajuan,jcperez,jcano,aperez,evidal,fcn}@iti.upv.es

Abstract

The main purpose of this work is to provide a qualitative description of the current research area on OCR carried out by the PRHLT-ITI/DSIC/DISCA research group. First, different preprocessing and features extraction methods are briefly described: tangent vectors based methods, local features extraction and others classical methods. Then, two different approaches for OCR are also presented: the k-nn method with its fast search version based on KD-Tree and the mixtures of Bernoulli classifier. Finally, a real implementation is shown at the end of this work.

Keywords: OCR, HMM, Continuous Distance Transformation, Local Features, Bernoulli Mixtures, Nearest Neighbours.

1 Introduction

OCR is an active research area in the PRHLT-ITI/DSIC research group. It has been approached from many points of view. The main purpose of this work is to provide a general qualitative description of the current research area on OCR carried out by the group, starting with an explanation of the different methodologies used for preprocessing and feature extraction and followed by a short description of the employed classification methodologies. Finally, some implemented OCR applications are presented in the last section.

* Work supported by the *Agencia Valenciana de Ciencia y Tecnología (AVCiT)" under grant GRUPOS03/031* and the Spanish project TIRIG (TIC 2003-08496-C)

2 Preprocessing and features extraction methods

In this section we describe different preprocessing and features extraction methods used in OCR tasks.

2.1 Preprocessing and features extraction for Bernoulli mixtures

OCR based on Bernoulli mixtures has been used by the PRHLT-ITI/DSIC research group in different works [13, 20, 21]. In them, OCR is applied to *Indian Digits* (figure 1). The dataset used comprises the 10425 digit samples included in the non-touching part of the *Indian digits database* provided by CENPARMI [1]. Original digit samples are given as binary images of different sizes (minimal bounding boxes). To obtain properly normalised images, both in size and position, two simple preprocessing steps were applied. First, each digit image was pasted onto a square background whose centre was aligned with the digit centre of mass. This square background was a white image large enough (64×64) to accommodate most samples though, in some cases, larger background images were required. Second, given a size S, each digit image was subsampled into $S \times S$ pixels, from which its corresponding binary vector of dimension $D = S^2$ was built. Figure 1 shows one preprocessed example of each Indian digit ($S = 30$).

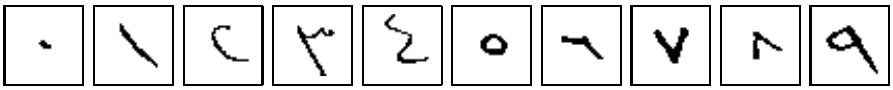


Figure 1: 30×30 examples of each Indian digit.

2.2 An off-line HMM-based OCR system for isolated handwritten lowercase letters

As result of the PRHLT Group's initial research work on the application of standard *continuous speech recognition* (CSR) technology in the *Off-line handwriting recognition* area, a *(continuous density) hidden Markov models* (HMM)-based OCR prototype-system for isolated handwritten character classification was implemented. It has been focused specially on finding adequate preprocessing and feature extraction methods for being used with (one-dimensional) HMM modelling.

A close attention has been paid to the recognition of *individual, isolated characters*, mainly for guiding system design when more complex tasks were confronted. Good results comparable with state-of-the-art results on the set of lowers in the

NIST Special Database 3 [14] have been reported in [11]. Finally it is also worth to mention that letter HMM training and recognition processes were both done on the basis of the well-known and widely available *standard Hidden Markov Model Toolkit (HTK)* for CSR [31].

Preprocessing and feature extraction are explained along with a brief description about how individual characters are modelled using HMMs as it follows.

Given a binary image of a character, the first step of our preprocessing module removes "specks of dust" (isolated and small connected components of black pixels) from the image and then fits a minimal bounding box to its remaining black pixels. The second step performs slant correction on the resulting cropped image using the method described in [30]. Preprocessing ends with a third step that computes the vertical density histogram (number of black pixels in each row) and smoothes it by merging consecutive rows with low density and replicating rows with high density. This has the effect of reducing the size of ascenders and descenders, which it is thought to be of help for our HMM-based character modelling approach. Figure 2 illustrates the second and third steps.

Given a preprocessed image, feature extraction transforms it into a *sequence of feature vectors*. To do this, the preprocessed image is first divided into a 24-rows grid of squared cells (a *vertical resolution* of 1/24 has been chosen after the results reported in [15]). Then each cell is characterised by the following features: *normalised grey level*, *horizontal grey-level derivative* and *vertical grey-level derivative*. Columns of cells or *frames* are processed from left to right and a feature vector is constructed for each frame by stacking the features computed in its constituent cells (see fig. 2).

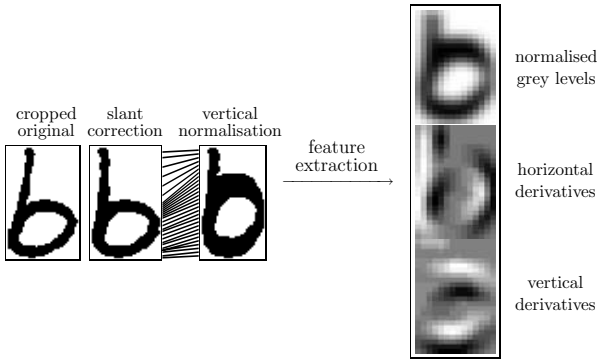


Figure 2: Preprocessing and feature extraction example.

Individual *characters* are modelled by *continuous density left-to-right hidden Markov models* (HMM), similar to those used in CSR [19] (see fig. 3). Basically,

each character HMM is a stochastic finite-state device aimed at modelling the succession, along the horizontal axis, of (vertical) feature vectors which are extracted from instances of this character. It is assumed that each HMM state generates feature vectors following an adequate parametric probabilistic law; typically, a *mixture of Gaussian densities*. The required number of densities in the mixture depends, along with many other factors, on the "vertical variability" typically associated with each state. This number needs to be empirically tuned in each task. On the other hand, the number of states that is adequate to model a certain character or character set depends on the underlying "horizontal variability". For instance, to ideally model a capital "E" character, only two states might be enough (one to model the vertical bar and the other for the three horizontal strokes), while three states may be more adequate to model a capital "H" (one for the left vertical bar, another for the horizontal stroke and the last one for the right vertical bar). The most appropriate number states for a given task also depends of the amount of training data which is available to train model parameters. So, the exact number of states to be adopted needs some empirical tuning in each practical situation. This training process is carried out using a well known instance of the EM algorithm called *backward-forward or Baum-Welch re-estimation* [19].

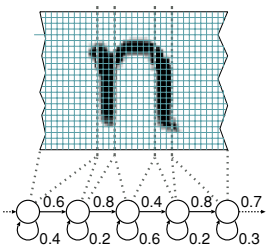


Figure 3: Example of the structure of a character left-to-right hidden Markov model.

2.3 The continuous distance transformation

Obtaining feature maps from images, where the distance relationships among their pixels are taken into account is the goal of a well-known technique usually referred to as *Distance Transformation* or DT [28]. The Distance Transformation is traditionally defined as an operation that transforms a binary image consisting of feature and non-feature pixels into a distance map, where all non-feature pixels have a value corresponding to the distance (any suitable distance function on the plane) to the nearest feature pixel [8]. Unfortunately, binarisation is a necessary step in order to compute the classical Distance Transforms from continuous-valued images, causing a loss of information.

A generalisation of the DT, the Continuous Distance Transformation (CDT), was presented as a technique to compute distance maps from continuous-valued images [3, 6]. Applicable to gray-level images, the CDT technique avoids binarisation process and make use of the whole information content of the original range of representation.

Taking the definition of Distance Transformation as a basis, an item (i, j) of a "Distance Map to the Nearest White Pixel" holds the distance from pixel (i, j) on the image to the nearest white pixel. Note that this value can be interpreted as the number of fringes expanded from (i, j) until the first fringe holding a white pixel is reached, where a "fringe" is defined as the set of pixels that are at the same distance of (i, j).

A parallelism between a distance map of binary images and one whose pixel values are defined in the gray-scale domain [0..MaxBright] implies the replacement of the "white pixel" concept by the "maximum bright value" and actions as "find the nearest white pixel" by "accumulate a maximum bright value on an expanding neighbourhood". Moreover, the value of an item on the continuous distance map is a function of the pixel value itself, as well as, of the number of fringes expanded until an accumulated bright value reaches a threshold according to a certain criteria of bright value accumulation, which is applied to the pixels belonging to each fringe analysed. Then, the concept of "distance to the nearest white pixel" is substituted by the concept of "distance from a pixel to the limit of their area of brightness saturation".

Two types of CDT-based maps can be defined: Continuous Distance Map to Brightness Saturation (CDTB), or generically, *Distance Map to Direct Saturation* (Θ^D), and Continuous Distance Map to Darkness Saturation (CDTD), or generically, *Distance Map to Reverse Saturation* (Θ^R), depending on if a maximum value of bright intensity or a maximum value of reverse bright intensity is accumulated, respectively. Both maps provide distinct information about a point and its surrounding area. In [3, 6], detailed descriptions of these concepts are presented. Given an image, either a Θ^D map or a Θ^R map are more or less descriptive depending on its brightness distribution. Figure 4 shows both CDT maps and both DT maps obtained from a character image. The cost of a CDT map computation is in $\Omega(m^2 \times n^2)$ for an image of $m \times n$ pixels, but, in practice, it is much lower.

In a vectorial classifier, a number of dissimilarity and metric measures can be used over a set of extracted features from objects. In that context, several distance and dissimilarity measures based on the CDT can be used to take advantage of the full possibilities of the representation obtained. The well-know Minkowski metrics (L_p-norms) can be computed over either Θ^D and/or Θ^R maps. Furthermore, the *Continuous Pixel Distances* (PDL_p) –also named Generalised Pixel Distances–, is a

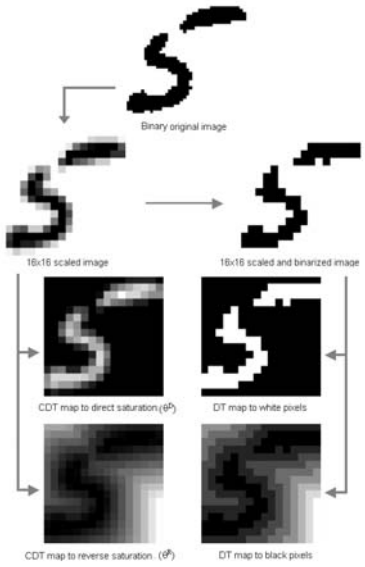


Figure 4: On the top, a 40x40 pixels binary digit. In the second line on the left, the gray-scale image resulting from scaling into 16x16 pixels and, on the right the corresponding 16x16 binarized image. Below the gray-scale image of the Θ^D and Θ^R maps are shown. Below the binarized image, their corresponding DT maps are shown. CDT maps show a wider range of gray values than DT maps because they contain more information.

family of specific CDT-based dissimilarity measures. They are based in the following concept of similarity between images: two images with values in the gray scale are more similar if the values of a pixel (i,j) are coincident or, otherwise, their respective neighbourhoods are similar. Taking into account that both Θ^D and Θ^R maps describe the neighbourhood of an image, the following expression computes the PDL_p distances between two continuous-valued images X and Y of $m \times n$ pixels defined in [0..MaxScale]

$$\mathrm{PDL}_p(X,Y) \quad = \quad \sum_{i=1}^{m}\sum_{j=1}^{n} \omega(i,j)\,(L_p(\Theta_X^D(i,j),\Theta_Y^D(i,j)) + L_p(\Theta_X^R(i,j),\Theta_Y^R(i,j)))$$

where

$$\omega(i,j) = \frac{|X(i,j) - Y(i,j)|}{\mathrm{MaxScale}}$$

The weight of the neighbourhood in the former expression can be tuned by the exponent p, and it is more significant as the differences between maps are higher.

Scaling character images is very common as a pre-process in OCR systems. The Continuous Distance Transformation used in conjunction with a k-nn classifier has been shown to provide good results in the task of handwriting character recognition [2, 6]. Table 1 shows some results using the SD3 database [16] tested on some CDT-based measures and the Euclidean distance. In all cases, the results show better performances of the CDT-based measures compared to the Euclidean distance.

Table 1: Error rates and confidence intervals (95%) obtained for some CDT-based measures. The best k value and the best grid are shown.

Distances	Grid	Lower-case		Upper-case		Digits	
ED	8x8	10.67	9.83 11.56	5.59	4.97 6.26	0.84	0.73 0.97
L5BD	28x28	9.08	8.30 9.91	3.66	3.15 4.21	0.81	0.70 0.93
PDL3	28x28	8.76	7.99 9.57	3.60	3.09 4.15	0.68	0.57 0.79

2.4 Local features

In a classical classifier, each object is represented by a feature vector, and a discrimination rule is applied to classify a test vector that also represents one object. Local representation, however, implies that each image is scanned to compute many feature vectors. Each of them could be classified into a different class (for instance using nearest neighbours), and therefore a decision scheme is required to finally decide a single class for a test image.

So, preprocessing method must be a quiet different when we use local features. We used local features in the OCR of digits [22] and is now been used in the OCR of Indian digits. Many local representations have been proposed. In our works, each image is represented by several (possibly overlapping) square windows of size $w \times w$, which correspond to a set of "local appearances" (figure 5).

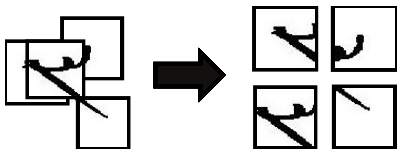


Figure 5: Example of four local features extracted from an image of an Indian handwritten digit.

To obtain the local feature vectors from an image, a selection of windows with highly relevant and discriminative content is needed. Although a number of methods

exist to detect such windows, most of them are not appropriate for handwritten images or they are computationally too expensive. In our works, the pixels black value is used as selection criterion as they are selected in order to determinate windows centers. For each of the selected pixels, a w^2-dimensional vector of grey values is first obtained in the preprocessed image by application of a $w \times w$ window around it. The dimension of the resulting vectors is then reduced from w^2 to 30 using *Principal Component Analysis* (PCA), thus obtaining a compact local representation of a region of the image. This is illustrated in figure 6.

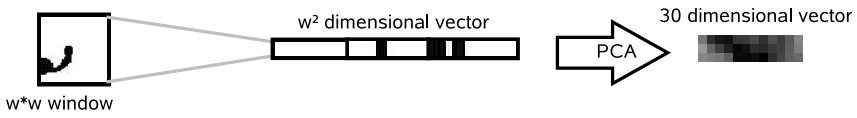


Figure 6: Feature extraction process.

3 Classification methods

In this section, several works related to OCR classifiers are presented. These works are based on two different approaches: the k Nearest Neighbours rule and the mixtures of Bernoulli classifier.

3.1 *k*-nearest neighbours

The k Nearest Neighbours (k-nn) Rule is a classical statistical method which offers consistently good results as well as it shows certain theoretical properties related to the expected error. The basic k-nn is a memory-based classifier which uses every stored prototype to be compared to the test observations, hence, it can benefit from the sample diversity coming from very large training datasets.

3.1.1 Fast and accurate handwritten character recognition using approximate nearest neighbours search on large databases

A number of studies [17, 29], have shown the power of k-nearest neighbour classifiers (k-nn) using large databases for character recognition. In those works, the error rate is found to decrease consistently as the size of the database increases. Unfortunately, a large database implies large search times if an exhaustive search algorithm is used. However, fast approximate nearest neighbours search algorithms on large databases are shown to provide high accuracies, similar to those of exact nearest neighbour search. Most of them are based on tree structures [7, 12]. Other ones are based on projections and space filling curves (SPFC) [18, 23, 24].

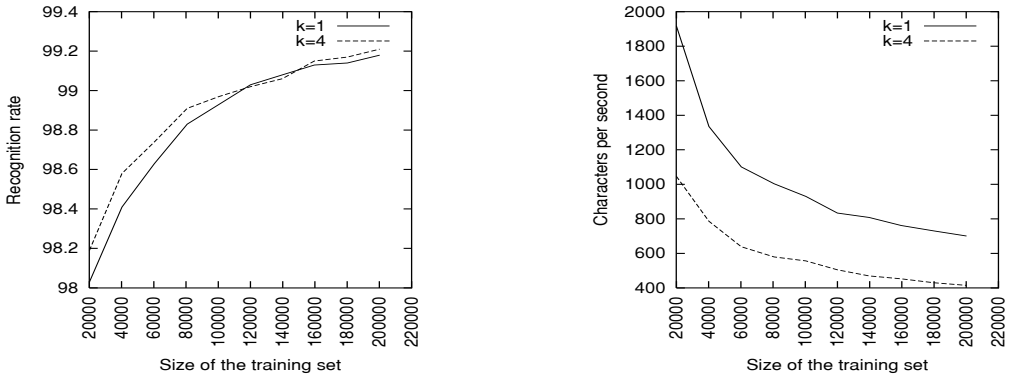


Figure 7: Recognition at zero percent rejection (left), and number of searches per second (right) for two different values of k, an approximate search parameter $\epsilon = 1.5$ and increasing training set sizes. Throughputs measured on a PentiumII - 450Mhz running Linux 2.2.9 not including preprocessing time of the test character.

In our work [26], experiments using fast and approximate SPFC and kd-trees algorithms were made over the SD3 database [16]. Such a experiments demonstrated that when applied to an OCR task, character recognition on large databases can be reached at a fraction of the computational cost from the exhaustive search. The improvements that can be expected using *kd*-trees search from training sets of increasing size, in terms of results and recognition speeds, are shown in Figure 7.

Given the slow increase of the search times incurred when the database grows, an interesting approach to improve the accuracy, keeping at the same time high recognition speeds, was to insert new prototypes into the training set [17]. Deformations based on slant were applied to the training characters and inserted in a new larger training set. The recognition rate using 4 slant angles to obtain a training set of 1,000,965 digits (including the 200,193 original ones) improved to 99.43%, from 99.21%, thus cutting the error rate by more than one fourth, in the test on SD3 digits [16], with k=4 and $\epsilon = 1.5$. The search time increased from 2.4 ms/char to 4.5 ms/char. The error rates and search times presented in [26] prove that k-nn can be a practical technique for a character recognition task.

3.1.2 Training set expansion in handwritten character recognition

Approximate nearest neighbours search in large databases can be successfully used in an OCR task, and significant performance improvements were obtained by simply increasing the size of the training set [26]. In our work [9], a process of expansion of the training set by synthetic generation of handwritten uppercase letters via defor-

mations of natural images is tested in combination with an approximate $k-$Nearest Neighbour ($k-$nn) classifier.

Four simple kinds of image transformations were tested (Figure 8): slant and shrink, to cope with geometric distortions of the writing, and erosion and dilation to account for different writing implements, acquisition conditions, etc. The transformations were first tested separately and then the one offering the best results (slant) was applied first, expanding the training set so that the rest of transformations were incrementally applied to the original plus the slanted characters.

Figure 8: Families of transformations tested

The accuracy improvement achieved by artificially expanding a core database of images is comparable to using extra real data. Another experiment was carried out, where a large locally acquired real database was used, allowing increasingly large training sets up to 674265 images, equivalent to the size of the previous artificially expanded database.

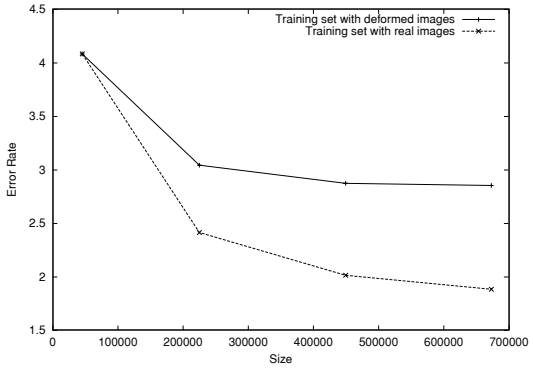


Figure 9: Comparison of error rates of a $k-$nn classifier for increasingly large training sets composed of real-only and real+synthetic images from the local database.

A core set of images from the local database was randomly selected. On the one hand, this core set was made larger by adding new real images from the rest of the local database, and on the other hand, the proposed expansion of the core using deformations was applied. In Figure 9, the results of this experiment on the local database using a $k-$nn classifier are shown. A recognition rate improvement is achieved in the classification of both real upper-case letters database and local

database (synthetically increased). However, this training set size increase does not seriously affect the processing time requirements of the recognition method. The results suggest that both approaches provide significant improvements.

3.1.3 Fast handwritten recognition using continuous distance transformation

The Continuous Distance Transformation used in conjunction with a k-nn classifier has been shown to provide good results in the task of character handwriting recognition [2]. Unfortunately, efficient techniques such as kd-tree search methods cannot be directly used in the case of certain dissimilarity measures like the CDT-based distance functions.

In our work [5], the problem of the computational complexity reduction associated to a k-nn classifier using complex distance functions was approached in a simple way: In a first step, fast search using kd-trees is applied to a test observation in order to get a number k' of nearest prototypes. Secondly, an exhaustive search of the k nearest neighbours, $k < k'$, among the k' pre-selected prototypes using specific features and distance functions is carried out to assess better performances. Notice that the space of features can be different each search. The computational cost of this combined methodology depends on k' and, in practice, is significantly lower than that of exhaustive search over the whole training set.

Table 2: Error rates of the three methods for k'=100, k=3, and ϵ=1.5 using the L6D, L9BD and PDL5 pre-selected CDT distance functions

Method \ Distance	L6D	L9BD	PDL5
Exhaustive CDT	4.17	3.93	3.69 (287 ms/char)
kd-tree (CDTD) & CDT	4.17	3.93	3.65 (7.78 ms/char)
kd-tree (image)	6.05 (3.69 ms/char)		
kd-tree (CDTD)	4.99 (3.17 ms/char)		

For handwritten character classification we tested some of the CDT-based distance functions. Table 2 shows the experimental results obtained using the SD3 upper-case database [16] for: 1st) exhaustive k-nn search using CDT distance functions, 2nd) the proposed methodology, 3rd) approximate search in kd-tree over the pixel feature space, and 4th) approximate search in kd-tree over the CDTD map feature space respectively. The recognition rates achieved have no significant differences with those found in an exhaustive k-nn classification using CDT distance

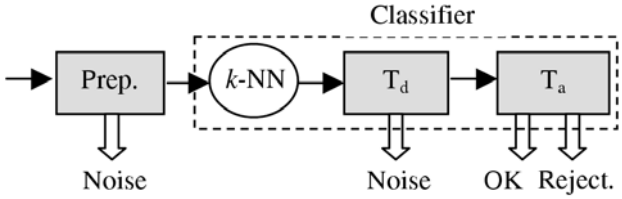


Figure 10: Stages of the proposed system.

functions, with a very important temporal cost reduction.

3.1.4 Rejection strategies and confidence measures for a $k-$nn classifier in an OCR task

In handwritten character recognition, the rejection of extraneous patterns, like image noise, strokes or corrections, can improve significantly the practical usefulness of a system. Confidence measures can be defined from the *a posteriori* probability provided by the *k-nn* classifier. However, a completely unrecognisable symbol, a crossing-out or a very noisy pattern can be classified with a high confidence when all or most of its neighbours are from the same class, even if the distances involved are unusually large. In our work [4], a combination of two confidence measures defined for a k-nearest neighbours classifier was proposed to reject such outlier patterns.

Figure 10 shows the proposed system which use a heuristic pre-processing step previous to the classification, where some clearly noisy patterns are detected. However, the general problem of detecting those patterns has to be addressed in the classification phase. Thus, we defined a procedure to obtain a function $g(x)$ which allows to reject a pattern y having a $g(y)$ value over a pre-fixed *distance threshold* T_d. Then, the non-rejected remaining ones are classified into a regular class using a standard *ambiguity threshold* T_a based on the *a posteriori* probability provided by the *k-nn* classifier.

Our approach is based on considering the need of a confidence measure representing the probability that a pattern is a character *of any class*. A natural way to address this problem is to estimate the probability under the p.d.f. of all classes, $\hat{p}(x) = \sum_i \hat{p}(x \mid \omega_i)$ [27] and the direct sum of the distances to the k nearest neighbours of x, $g'(x)$, can be regarded as an "inverse confidence measure" related to that estimation:

$$g'(x) = \sum_{j=1}^{k} d(x, y_j)$$

Obviously, this function is not a p.d.f. and, from a practical viewpoint, the fact that the range of values obtained is not bounded, but depends on the magnitudes of the distances, is a significant problem. Establishing a consistent *distance threshold* T_d for different training sets, space dimensionalities or distance measures becomes difficult and inconvenient. To normalise $g'(x)$, a suitable reference has to be used. Thus, the distribution function of the values of $g'(x)$, for x in a representative sample of observations is proposed:

$$g(x) = F\left(g'(x)\right)$$

Finally, we implemented an estimation of this distribution function with an accumulated histogram of the values of $g'(x)$ from the prototypes in the training set, using a leaving-one-out technique. Thus, in the test phase, the rule $g(y) \geq T_d$ is applied to reject outlier patterns. Typical values for T_d should be: slightly below 1 when the existence of outliers in the training set is known or suspected, and 1 or slightly over 1 when the training set is known to be clean and representative of the whole population. In Table 3, the number of characters and abnormal patterns rejected at the pre-process and at the distance rejection stage are shown.

Table 3: Patterns rejected at pre-processing and in the distance rejection phase.

	Initial	Pre-process	$T_d = 0.97$
Letters	6273	4 (0.06%)	207 (3.29%)
Abnormal	824	34 (4.13%)	462 (56.1%)

Experiments to compare the performances of a system with ambiguity rejection but no distance rejection option (SYS-A) and a system using both rejection tests (SYS-B) were made. The systems were tested using different percentages of abnormal patterns along with regular characters. The results showed better for SYS-B when the number of abnormal patterns in the test set is higher than 5% or 10% at high rejection rates, common in practice, at the expense of a small loss of performance at low rejection rates when the system operates on "clean" test sets.

3.2 Bernoulli mixtures

Bernoulli mixtures is other classification method used in OCR task [13, 20, 21]. Mixture modelling is a popular approach for density estimation in both supervised and unsupervised pattern classification. On one hand, mixtures are flexible enough for finding an appropriate tradeoff between model complexity and the amount of training data available. Usually, model complexity is controlled by varying the number of mixture components while keeping the same (often simple) parametric

form for all components. On the other hand, maximum likelihood estimation of mixture parameters can be reliably accomplished by the well-known *Expectation-Maximisation (EM)* algorithm.

A (finite) mixture model consists of a number of *mixture components, I*. It generates a D-dimensional *sample* $\vec{x} = (x_1, \ldots, x_D)^t$ by first selecting the ith component with *prior probability* $p(i)$, and then generating $\vec{x}$ in accordance with the ith *component-conditional probability (density) function* $p(\vec{x} \,|\, i)$. The priors must satisfy the constraints:

$$\sum_{i=1}^{I} p(i) = 1 \quad \text{and} \qquad p(i) \geq 0 \quad (i = 1, \ldots, I). \tag{1}$$

The *posterior probability* of $\vec{x}$ being actually generated by the ith component can be calculated via the *Bayes' rule* as

$$p(i \,|\, \vec{x}) = \frac{p(i)\, p(\vec{x} \,|\, i)}{p(\vec{x})} \tag{2}$$

where

$$p(\vec{x}) = \sum_{i=1}^{I} p(i)\, p(\vec{x} \,|\, i) \tag{3}$$

is the *(unconditional) mixture probability (density) function*. A Bernoulli mixture model is a particular case of (3) in which each component i has a D-dimensional Bernoulli probability function governed by its own vector of parameters or *prototype* $\vec{p}_i = (p_{i1}, \ldots, p_{iD})^t \in [0,1]^D$,

$$p(\vec{x} \,|\, i) = \prod_{d=1}^{D} p_{id}^{x_d} \, (1 - p_{id})^{1 - x_d} \tag{4}$$

Consider an arbitrary component $p(\vec{x} \,|\, i)$. It identifies a certain *subclass* of binary vectors "resembling" its parameter vector or *prototype* $\vec{p}_i$. In fact, each p_{id} is the probability of bit x_d to be one, whereas $1 - p_{id}$ is the opposite. Equation (4) is just the product of independent, unidimensional Bernoulli probability functions. Therefore, a single multivariate Bernoulli component can not capture any kind of dependencies or correlations between individual bits. As with other types of mixtures, this is implicitly done by mixing several components in the right proportions.

Also as with other types of mixtures, Bernoulli mixtures can be used as class-conditional models in supervised classification tasks. Let C denote the number of supervised classes. Assume that, for each supervised class c, we know its prior

$p(c)$ and its class-conditional probability function $p(\vec{x}\,|\,c)$, which is a mixture of I_c Bernoulli components,

$$p(\vec{x}\,|\,c) = \sum_{i=1}^{I_c} p(i\,|\,c)\, p(\vec{x}\,|\,c,i) \tag{5}$$

Then, the optimal Bayes decision rule is to assign each pattern vector $\vec{x}$ to a class $c^*(\vec{x})$ giving maximum a posteriori probability:

$$\begin{aligned} c^*(\vec{x}) &= \arg\max_c p(c\,|\,\vec{x}) && (6)\\ &= \arg\max_c \Big(p(c)\, p(\vec{x}\,|\,c) \Big) && (7)\\ &= \arg\max_c \Big(\log p(c) + \log p(\vec{x}\,|\,c) \Big) && (8)\\ &= \arg\max_c \Big(\log p(c) + \log \sum_{i=1}^{I_c} p(i\,|\,c) p(\vec{x}\,|\,c,i) \Big) && (9) \end{aligned}$$

As it is said above. maximum likelihood estimation of mixture parameters can be reliably accomplished by the well-known *Expectation-Maximisation (EM)* algorithm.

Let $X = \{\vec{x}_1, \ldots, \vec{x}_N\}$ be a set of samples available to learn a Bernoulli mixture model. This is a statistical parameter estimation problem since the mixture is a probability function of known functional form, and all that is unknown is a parameter vector including the priors and component prototypes:

$$\vec{\Theta} = (p(1), \ldots, p(I), \vec{p}_1, \ldots, \vec{p}_I)^t. \tag{10}$$

Here we are excluding the number of components from the estimation problem, as it is a crucial parameter for controlling model complexit. Following the maximum likelihood principle, the best parameter values maximise the log-likelihood function of $\vec{\Theta}$,

$$\mathcal{L}(\vec{\Theta}\,|\,X) = \sum_{n=1}^{N} \log \left(\sum_{i=1}^{I} p(i)\, p(\vec{x}_n\,|\,i) \right). \tag{11}$$

In order to find these optimal values, it is useful to think of each sample $\vec{x}_n$ as an *incomplete* component-labelled sample, which can be completed by an indicator vector $\vec{z}_n = (z_{n1}, \ldots, z_{nI})^t$ with 1 in the position corresponding to the component generating $\vec{x}_n$ and zeros elsewhere. In doing so, a complete version of the log-likelihood function (11) can be stated as

$$\mathcal{L}_C(\vec{\Theta}|X, Z) = \sum_{n=1}^{N} \sum_{i=1}^{I} z_{ni} \left(\log p(i) + \log p(\vec{x}_n|i) \right), \tag{12}$$

where $Z = \{\vec{z}_1, \ldots, \vec{z}_N\}$ is the so-called *missing* data.

The form of the log-likelihood function given in (12) is generally preferred because it makes available the well-known *EM* optimisation algorithm (for finite mixtures). This algorithm proceeds iteratively in two steps. The E(xpectation) step computes the expected value of the missing data given the incomplete data and the current parameters. The M(aximisation) step finds the parameter values which maximise (12), on the basis of the missing data estimated in the E step. In our case, the E step replaces each z_{ni} by the posterior probability of $\vec{x}_n$ being actually generated by the ith component,

$$z_{ni} = \frac{p(i)\, p(\vec{x}_n \,|\, i)}{\sum_{i'=1}^{I} p(i')\, p(\vec{x}_n \,|\, i')} \qquad \begin{pmatrix} n = 1, \ldots, N \\ i = 1, \ldots, I \end{pmatrix}, \tag{13}$$

while the M step finds the maximum likelihood estimates for the priors,

$$p(i) = \frac{1}{N} \sum_{n=1}^{N} z_{ni} \qquad (i = 1, \ldots, I), \tag{14}$$

and the component prototypes,

$$\vec{p}_i = \frac{1}{\sum_{n=1}^{N} z_{ni}} \sum_{n=1}^{N} z_{ni} \vec{x}_n \qquad (i = 1, \ldots, I). \tag{15}$$

To start the EM algorithm, initial values for the parameters are required. To do this, it is recommended to avoid "pathological" points in the parameter space such as those touching parameter boundaries and those in which the same prototype is used for all components. Provided that a non-pathological starting point is used, each iteration is guaranteed not to decrease the log-likelihood function and the algorithm is guaranteed to converge to a proper stationary point (local maximum). Also, for the sake of robustness, it is important to introduce some sort of model smoothing.

Although most research in mixture modelling has focused on mixtures for continuous data, there are many pattern recognition tasks for which binary or discrete mixture models are better suited. For instance, Bernoulli mixtures has been used in the OCR of Indian digits [13, 20, 21] (see figure 1).

4 OCR post-proces

In [25], stochastic error-correcting parsing is proposed as a powerful and flexible method to post-process the results of an optical character recogniser (OCR). Deterministic and non-deterministic approaches are possible under the proposed setting.

The basic units of the model can be words or complete sentences, and the lexicons or the language databases can be simple enumerations or may convey probabilistic information from the application domain.

5 A real OCR application: license plates recognition

A robust method for plate segmentation in a License Plate Recognition (LPR) system is presented in [10]. It is designed to work in a wide range of acquisition conditions, including unrestricted scene environments, light, perspective and camera-to-car distance. Although a novel text-region segmentation technique was applied to a very specific problem, it is extensible to more general contexts, like difficult text segmentation tasks dealing with natural images.

In this task, due the nature of images (unrestricted context), a segmentation method capable of generating various hypothesis for each image was implemented in order to prevent the loss of any possible license plate region. Accordingly, a subsequent recognition phase that filters the final results without discarding beforehand any reasonable segmentation hypothesis was designed to obtain the plate identifier.

Figure 11: Example image of the test set for the license plate recognition task

The overall process of an image provided de user with a set of plates identifiers and corresponding confidence measures, as shown in the example, 12. Each one of the hypotheses could come from differnet areas of the original image (11) or from different image scales.

References

[1] Y. Al-Ohali, M. Cheriet, and C. Suen. Databases for recognition of handwritten Arabic cheques. *Pattern Recognition*, 36:111–121, 2003.

Figure 12: Set of pairs plate identifier hypothesis & corresponding confidence

[2] Arlandis J., Perez-Cortes J.C. and Llobet R., *Handwritten Character Recognition Using Continuos Distance Transformation*, Proceedings of the 15th. International Conference on Pattern Recognition **1** (2000) 940–943.

[3] Arlandis J. and Perez-Cortes J.C., *The Continuos Distance Transformation: A Generalization of the Distance Transformation for Continuos-valued Images*, Pattern Recognition and Applications **56** (2000) 89-98.

[4] J. Arlandis, J.C. Perez-Cortes and J. Cano, *Rejection Strategies and Confidence Measures for a k-nn Classifier in an OCR Task*, 16th. International Conference on Pattern Recognition, volume 1,576–579,Quebeq,Canada, IEEE Computer Society,2002.

[5] J. Arlandis and J.C. Perez, *Fast Handwritten Recognition Using Continuous Distance Transformation*, Progress in Pattern Recognition Speech and Image Analysis, November 2003, Editors A. Sanfeliu and J. Ruiz-Shulcloper, LNCS 2905, Springer, pages 400–407.

[6] Joaquim Arlandis, *La transformació contínua de la distància. Estudi i aplicació a un sistema OCR*, PhD thesis,Dep. Informàtica de Sistemes i Computadors, Universitat Politècnica de València,València (Spain), mar 2004.

[7] Arya S., Mount D.M., Netanyahu N.S., Silverman R. and Wu A., *An optimal algorithm for approximate nearest neighbor searching*, Journal of the ACM **45** (1998) 891-923.

[8] Borgefors C., *A New Distance Transformation Approximating the Euclidean Distance*, Proceedings of the 8th. International Conference on Pattern Recognition, **1** (1986) 336–338.

[9] J. Cano, J.C. Perez-Cortes, J. Arlandis and R. Llobet, *Training Set Expansion in Handwritten Character Recognition*, International Workshop on Statistical Pattern Recognition SPR-2002, Windsor (Ontario, Canada),548-556,2002, LNCS 2396.

[10] J. Cano and J.C. Perez-Cortes, *Vehicle License Plate Segmentation In Natural Images*, Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), Puerto de Andratx (Mallorca, Spain),142-149,2003, volume 1.

[11] J. Doménech, A. H. Toselli, A. Juan, E. Vidal, and F. Casacuberta. An off-line HTK-based OCR system for isolated handwritten lowercase letters. In *Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volume II, pages 49–54, Benicàssim (Spain), May 2001.

[12] Friedman J.H., Bentley J.L. and Finkel R.A., *An algorithm finding best matches in logarithmic expected time*, ACM Trans. Math. Software **3** (209–226).

[13] José García-Hernández, Vicent Alabau, Alfons Juan and Enrique Vidal, *Bernoulli mixture-based classification*, Proc. of the LEARNING04, 2004

[14] M. D. Garris and R. A. Wilkinson. Handwritten segmented characters database. Technical Report Special Database 3, NIST, February 1992.

[15] J.González et al. Off-line Recognition of Syntax-Constrained Cursive Handwritten Text. In *Proc. of the S+SSPR 2000*, pages 143–153, Alicante, 2000.

[16] P.J. Grother, *NIST Special Database 19. Handwprinted Forms and characters Database*, Technical Report,National Institute of Satandards and Technology, Maryland (USA), 1995.

[17] Ha T.M. and Bunke H., *Off-Line, Handwritten Numeral Recognition by Perturbation Method*, IEEE Trans. on PAMI, volume 19, number 5, 1997,May,535-539.

[18] H.V. Jagadish, *Linear clustering of objects with multiples atributes*, International Conference on Management of Data,332–342,1990.

[19] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[20] Alfons Juan, José García-Hernández and Enrique Vidal. *EM Initialisation for Bernoulli Mixture Learning*, Proc. of the SSPR-SPR04, LNCS 3138, 635–643, 2004.

[21] Alfons Juan and Enrique Vidal, *Bernoulli mixture models for binary images*, Proc. of the 17th Int. Conf. on Pattern Recognition (ICPR 2004), 2004, volume 3,Cambridge (UK).

[22] D. Keysers,d R. Paredes, H. Ney and E. Vidal, *Combination of Tangent Vectors and Local Representations for Handwritten Digit Recognition*, In SPR 2002.

[23] S. Nene and S. Nayar, *Closest point search in high dimensions*, Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition, volume 1,859–865,1996.

[24] Perez J.C. and Vidal E., *An Approximate Nearest Neighbours Search Algorithm Based on the Extended General Spacefilling Curves Heuristic*, Lecture Notes in Artificial Intelligence **1451** (1998) 697–706.

[25] Perez-Cortes J.C., Amengual J.C., Arlandis J. and Llobet R., *Stochastic Error Correcting Parsing for OCR Post-processing*, International Conference on Pattern Recognition ICPR-2000.

[26] Perez-Cortes J.C., Arlandis J. and Llobet R., *Fast and Accurate Handwritten Character Recognition using Approximate Nearest Neighbours Search on Large Databases*, Lecture Notes in Artificial Intelligence, volume 1876, 767-776, 2000, Alicante (Spain)

[27] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press,Cambridge,1996,ISBN 0-521-46086-7.

[28] Rosenfeld A. and Pfaltz J.L., *Sequential Operations in Digital Picture Processing*, Journal of the ACM **13(4)** (1966) 471–494.

[29] Smith S.J., Bourgoin M.O., Sims, K. and Voorhees H.L. *Handwritten Character Classification using Nearest Neighbor in Large Databases*, IEEE Trans. on PAMI,volume 16,Number 9, 1994, September, 915-919.

[30] B. Yanikoglu and P.A. Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, 31(12):1825–1833, 1998.

[31] S.J. Young, P. C. Woodland, and W.J. Byrne. HTK: Hidden Markov Model Toolkit V1.5. Technical report, Entropic Research Laboratories Inc., 1993.

Some improvements on NN based classifiers in metric spaces

Francisco Moreno-Seco, Luisa Micó, Jose Oncina
Dept. Lenguajes y Sistemas Informáticos
Universidad de Alicante, E-03071 Alicante, Spain
{paco,mico,oncina}@dlsi.ua.es

Abstract

The nearest neighbour (NN) and k-nearest neighbour (k-NN) classification rules have been widely used in Pattern Recognition due to its simplicity and good behaviour. Exhaustive nearest neighbour search may become unpractical when facing large training sets, high dimensional data or expensive dissimilarity measures (distances). During the last years a lot of fast NN search algorithms have been developed to overcome those problems, and many of them are based on traversing a data structure (usually a tree) testing several candidates until the nearest neighbour is found.

When these algorithms are extended to find the k nearest neighbours, the classification time increases with the value of k. In this paper we propose a new classification rule that makes use of the prototypes that are selected by these algorithms in a 1-NN search as candidates to nearest neighbour. To illustrate the behaviour of this rule, several fast and widely known NN search algorithms have been extended with it, obtaining classification results similar to those of a k-NN ($k > 1$) classifier without the extra computational overhead. Also, previous work on approximate NN search for vector spaces has been extended to algorithms suitable for general metric spaces, and has been combined with the new classification rule.

Keywords Nearest Neighbour, Classification Rule, Approximate Search

1 Introduction

Given a set P of prototypes, where each $p \in P$ belongs to one of a finite set of classes C, the nearest neighbour (NN) rule classifies an unknown sample into the class of its nearest neighbour in P according to some similarity measure (a *distance*). Despite its simplicity, the classification accuracy is usually enough for many Pattern Recognition tasks. However, some tasks may require lower classification error rates, and usually the k-NN rule [1] is used as a generalisation of the NN rule. The k-NN classification rule is also simple: find the k nearest neighbours of the sample and

```
    d_nn := ∞
    do until the training set P is empty
        p_i := argmin_{p∈P} Aprox(x, p)                    // Approximation
(a)     d := d(x, p)
        if d < d_nn then
                nn := p ; d_nn := d                        // Update the current NN
(b)             P := P − { q : q ∉ E(x, d_nn) }             // Elimination
        endif
    enddo
```

Figure 1: Approximation and elimination framework.

classify it by voting with the classes of the k nearest neighbours, i.e., assign the majority class to the sample.

Although initially used in Pattern Recognition, the NN rule has been also of interest for other fields such as data mining and information retrieval, which usually involves searching in very large databases and facing with high dimensionality data. Whenever the classification task requires large training sets, expensive distance measures or high dimensionality, the simple exhaustive search for the NN becomes unpractical. To overcome some of these problems, a large number of fast NN search algorithms [2, 3, 4, 5, 6, 7] have been developed. Many of these algorithms are suitable for any kind of prototype representation (vectors, strings, trees, ...) which allows to define a distance that holds the properties of a metric, that is, they do not assume that the prototype is a vector and thus they do not make use of the coordinates (see the work by Chávez et al. [8] for a review on NN search algorithms in metric spaces).

Most of these fast NN search algorithms may be easily extended to find the k-NN. However, the requirement of finding exactly the k-NN involves higher computing effort, and that effort increases with the value of k.

Several search algorithms can fit into an approximation and elimination framework [9], that can be formulated as in figure 1. The search process is seen as an iterative process: using an approximation function, a candidate to nearest neighbour is selected and its distance to the sample is computed. Then, if it is closer to the sample than the current NN, it becomes the current NN and the training set is pruned so that all the prototypes that are outside an hypersphere centered in the sample with radius d_{nn} (the distance to the current NN) are safely eliminated from the training set. The process continues until the training set is empty, and then the current NN will be the NN.

In this paper we propose a new classification rule that makes use of the prototypes that are selected in a standard 1-NN search as candidates to nearest neighbour by fast NN search algorithms. To illustrate the behaviour of this rule, several fast and widely known NN search algorithms have been extended with it, obtaining classification results similar to those of a k-NN ($k > 1$) classifier without the extra computational overhead. Also, previous work on approximate NN search for vector spaces has been extended to algorithms suitable for general metric spaces, and has been combined with the new classification rule.

The paper is structured as follows: the next section briefly describes the new classification rule, which is based on the approximation and elimination framework. Then, the approximate NN search for some (metric spaces) algorithms is outlined. The experiments section will show the results of the new rule when applied to various NN search algorithms in experiments with synthetic and real data, and also the results of combining the rule with approximate NN search. Finally, we will conclude and outline some future work.

2 The k-NSN classification rule

Many approximation and elimination search algorithms are based on the following idea: during preprocessing, a data structure is built to allow pruning of the training set. Then, during classification, a candidate to nearest neighbour is selected and stored, and its distance to the sample is computed. This distance is then used to prune the training set (using the data structure) and maybe to select a new candidate. This process ends when all the training set has been pruned or selected. Extending such an algorithm to find the k-NN is usually simple: each time a distance is computed (step (a) in figure 1), it is stored (along with the prototype) in a sorted array that holds the k-NN found so far. Then, the distance used to prune the training set is the distance to the kth nearest neighbour found so far, instead of the distance to the current NN (d_{nn} in step (b) of figure 1). This involves less pruning and more distances to compute, which derives in an additional computational overhead, always dependent on the value of k.

In this paper we propose a simple but powerful extension for any approximation and elimination based NN search algorithm: when looking for the nearest neighbour, each prototype selected and its distance to the sample are stored in a sorted array, as for the k-NN search. However, the distance used to prune the training set is d_{nn}, so that the number of distances (and thus the computational effort) is the same as for a standard NN search. The prototypes stored are called the k nearest selected neighbours (k-NSN), and they are not exactly the k-NN. When the search finishes,

the sample is classified by majority voting using these neighbours (which include the nearest neighbour), as in the k-NN rule.

This technique can be considered a new classification rule (the k-NSN rule) which requires very little computational effort over a NN search (storing the k nearest selected neighbours)[1], and, as we shall see in a following section, it achieves classification results very similar to those of the k-NN rule. Also, if this rule is applied to an exhaustive NN search it yields the k-NN rule. The rule raises up as an extension of previous work on the LAESA algorithm [10, 11]. Given that classification time does not (highly) depend on the value of k, one may increase k as desired to improve classification rates; however, as also happens with the k-NN rule, from a certain value of k the rates start to worsen.

3 Approximate NN search in metric spaces

There are a number of real tasks for which finding exactly the NN (even using a fast NN search algorithm) may become too slow; a number of approximate NN search algorithms [12, 13, 14, 15] have been proposed to face these tasks, yielding slightly worse classification rates but obtaining much lower classification times.

However, these algorithms are usually based on vector spaces of representation, and this feature limits its range of application in Pattern Recognition tasks. Moreover, in some real tasks where a string or tree represents an object (and thus usually the string or tree edit distance is used), classification times are much higher than in vector space tasks. In this work we have extended some ideas from previous works on approximate NN search in vector spaces to algorithms suitable for general metric spaces: Fukunaga and Narendra's [2], AESA [5], LAESA [10] and TLAESA [16].

In a widely known implementation of approximate NN search by Arya and Mount [12], a priority queue is used in a kd-tree to store the nodes which the search algorithm has still to visit. The key for the queue is some kind of lower bound of the distance from the node to the sample, and the node with the minimum key is the first to be extracted from the queue.

We have tested a similar idea with the Fukunaga and Narendra's algorithm, and with TLAESA (both tree-based algorithms): when a non-leaf node is visited, its children are stored in the queue using a key, m, which is a lower bound of the

[1] The simplest implementation is to insert the new pair distance/prototype in a sorted array of k pairs, if the distance is lower than the last one in the array. The extra time complexity over the NN search is $O(ck)$, where c is the number of computed distances. Although it is possible to reduce this time complexity with a heap, this overhead is almost negligible when compared to the overhead of computing c distances.

distance of the node to the sample[2]. In each step of the algorithm, the node with the minimum key m is extracted from the queue, and the algorithm finishes when the following condition holds:

$$m > d_{nn}$$

where d_{nn} is the distance of the current nearest neighbour to the sample. For an approximate search, the condition has been changed to:

$$m \cdot (1 + \epsilon) > d_{nn}$$

where ϵ is a parameter to tune the search: the higher the value of ϵ, the faster the search, but the higher the error rate. The optimum value of ϵ is a trade-off between classification time and allowable increase in the error rate, and should be determined for each classification task.

In the Fukunaga and Narendra's algorithm, any non-leaf node p has a representative M_p and a radius r_p. When a node is visited, the distances of the representatives of its children to the sample are computed and stored. Given a child p, the expression:

$$d(x, M_p) - R_p$$

is a (pessimistic) lower bound of the actual distance of the prototypes contained in p to the sample. Thus, if the child is not eliminated by the elimination condition of the algorithm (from which is derived the expression for the lower bound), it is stored in the queue along with the lower bound as the key.

The TLAESA algorithm does not compute any distances when visiting a non-leaf node; instead, it uses a lower bound of the distance from the representative M_p to the sample, $G[M_p]$, which is computed in the following way:

$$G[M_p] = \max_{b \in B} |d(x, b) - d(b, M_p)|$$

where B is a subset of prototypes called *base prototypes* (see [10, 16] for the details). The distances from each $b \in B$ to the sample are computed in the first step of the classification phase, and the distances $d(b, M_p)$ are computed (and stored) prior to classification, in a preprocessing step. The extension of this algorithm for approximate NN search has been done in a way very similar to that in the Fukunaga and Narendra's algorithm. The key for each node, which is also a lower bound of the distance of all the prototypes in the node to the sample, is:

$$G[M_p] - R_p$$

[2]In the work by Arya and Mount, only the unvisited child (in a binary tree like the kd-tree) is stored in the queue.

so that the lower bound $G[M_p]$ is used instead of $d(x, M_p)$.

The AESA and LAESA algorithms are not based on trees, and thus a different scheme has been used. Both algorithms also compute a lower bound of the distance of each prototype to the sample; in the case of LAESA, the lower bound is computed exactly as in the TLAESA algorithm (in fact, the TLAESA algorithm is derived from the LAESA algorithm). In the AESA algorithm, a lower bound of the distance is also computed, but in a slightly different way (as if all the prototypes were base prototypes). In each step of the search phase of the AESA or LAESA the algorithm, the prototype p whose lower bound $G[p]$ is the minimum is selected as a candidate to NN; whenever

$$G[p] > d_{nn}$$

the algorithm finishes. For approximate NN search, this condition has to be changed into this one:

$$G[p] \cdot (1 + \epsilon) > d_{nn}$$

No further changes are needed in both algorithms. This kind of approximate search with these two algorithms is very similar to the search using a certain looseness [17].

4 Experiments with the k-NSN rule

Several series of experiments have been performed in order to test the application of the k-NSN rule to various fast NN search algorithms (see table 1). All these algorithms fit in an approximation and elimination framework, and all are suited for general metric spaces except kd-tree, which requires point coordinates. The algorithms of AESA family (AESA, LAESA, and TLAESA) focus on reducing the number of distance computations, thus are best suitable for expensive distances. The vp-tree and GNAT were developed to face large training sets and/or high dimensionality of data, and thus the number of distance computations is important but it is not its main goal.

Two sets of experiments have been performed: first, a set of synthetic data experiments to test the performance of the rule in a widely known environment. Second, several tests have been performed with a real data task, human chromosome classification. In both cases the main goal was to study the error rates of these algorithms using the k-NSN rule and to compare them with the k-NN error rates.

4.1 Experiments with synthetic data

For these experiments we have generated Gaussian data from 8 classes of dimensions 10 and 20 using the algorithm for generating clustered data in [18]. Tests have been

ALGORITHM	Author(s)
kd-tree	Friedman *et al.*[3]
FN75	Fukunaga and Narendra [2]
vp-tree	Yianilos [4]
AESA	Vidal [5]
LAESA	Micó *et al.* [10]
TLAESA	Micó *et al.* [16]
GNAT	Brin [6]

Table 1: Fast NN search algorithms which have been extended with the k-NSN rule.

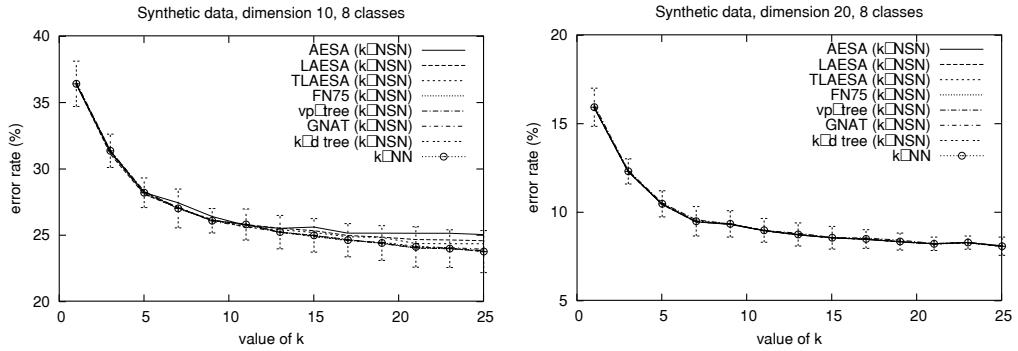


Figure 2: Comparison between k-NN and k-NSN classifiers, with synthetic data of dimensions 10 and 20. The error bars are plotted only for the k-NN rule and correspond to a 95% confidence interval.

performed using 10 differents pairs of training and test sets, with 4096 and 1024 prototypes respectively. The plots compare the error rate of the k-NSN and k-NN rules as the value of k increases (figure 2). These results show that k-NSN and k-NN error rates are very similar (and are almost the same for dimension 20), and are better than those of an NN classifier.

Although the definition of the k-NSN rule assures that the number of distance computations remains the same as in the NN rule, an experiment has been developed to verify it and also to show that in the k-NN rule the number of distances increase with the value of k. Figure 3 shows the results for the Fukunaga and Narendra's algorithm, named FN75 in the plots.[3]

[3]The results with all the other algorithms are similar and are not showed for brevity.

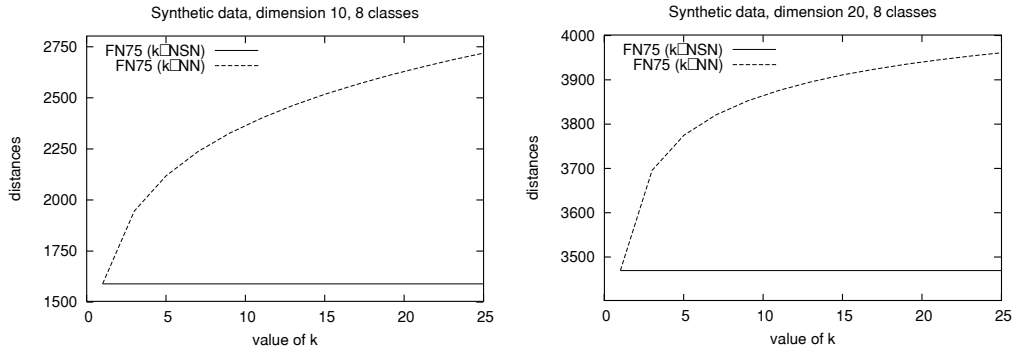


Figure 3: Average distance computations of k-NSN and k-NN rules using the Fukunaga and Narendra's algorithm (FN75), with synthetic data of dimensions 10 and 20.

4.2 Experiments with real data

The real data experiments have been developed for a human chromosome classification task [19, 20, 21]. The chromosome database contains 4400 samples coded as strings, and the edit or Levenshtein distance [22] has been used for this task; the kd-tree makes use of the coordinates of the prototypes, so it has not been tested with this database. The database is splitted into two sets of 2200 samples each, and two experiments have been performed using one of them for training and the other one for test. Figure 4 shows the average error rates of k-NN and k-NSN classifiers as the value of k increases. There is a parameter for LAESA and TLAESA (see [10, 16] for more details), the number of *base prototypes*, which has been set to 40.

The average number of distance computations and classification times are plotted in figures 5 and 6. Each individual plot compares k-NN and k-NSN values as the value of k increases. The results confirm that the number of distance computations and the average classification time per test sample of the k-NSN rule does not depend on the value of k; also, the plots show the main advantage of the k-NSN rule over the k-NN rule: whereas in the k-NN rule the time performance worses as the value of k increases, the k-NSN rule maintains the same time as the NN rule, while obtaining almost the same error rates than the k-NN rule.

4.3 How many of the k-NSN are among the k-NN?

The k-NSN rule obtains error rates very close to those of the k-NN, and one may think that this is due to the fact that many of the k-NSN are in fact among the

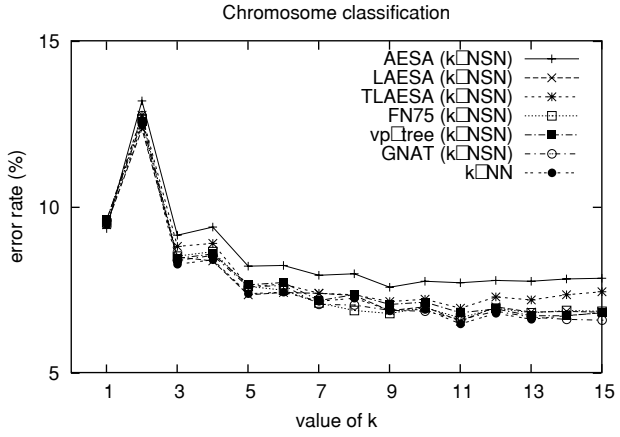


Figure 4: Error rate of k-NN and k-NSN rules in chromosome classification.

k-NN. Another possibility is that the k-NSN, even not matching exactly with the k-NN, are near at the same distance to the sample.

In order to study this question we have developed two experiments: in the first one, using synthetic data from varying dimensions (from 5 to 50) and taking two measures: the percentage of matching, that is, how many of the k-NSN are among the k-NN, and the relation between the distance of the last k-NSN and the last k-NN. The experiment has been repeated with 10 different pairs of training and test sets of 4096 and 1024 prototypes respectively. Figures 7 and 8 show a plot and a table with the results, which show that when dimension increases both the percentage of matching and the relation get close to the optimum (100% and 1). This is why the k-NSN error rates are almost the same as the k-NN rates when data dimension increases.

Our second experiment on this question was far more ambitious. We thought that if the percentage of matching clearly closes or reaches 100% when increasing the training set size, we could state that, in the limit (when the training set size closes infinite), the k-NSN match exactly the k-NN, and thus the k-NSN rule has the same statistical properties as the k-NN rule, i.e., that its error rate is bounded by as much 2 times the Bayes error [23]. The second experiment tries to see if this hypothesis holds. In this case, the training set size was varying from 1024 to 65536, with dimension 10, and the test set had 1024 prototypes. The results are plotted in figure 9, and they seem to prove that our hypothesis was not true.

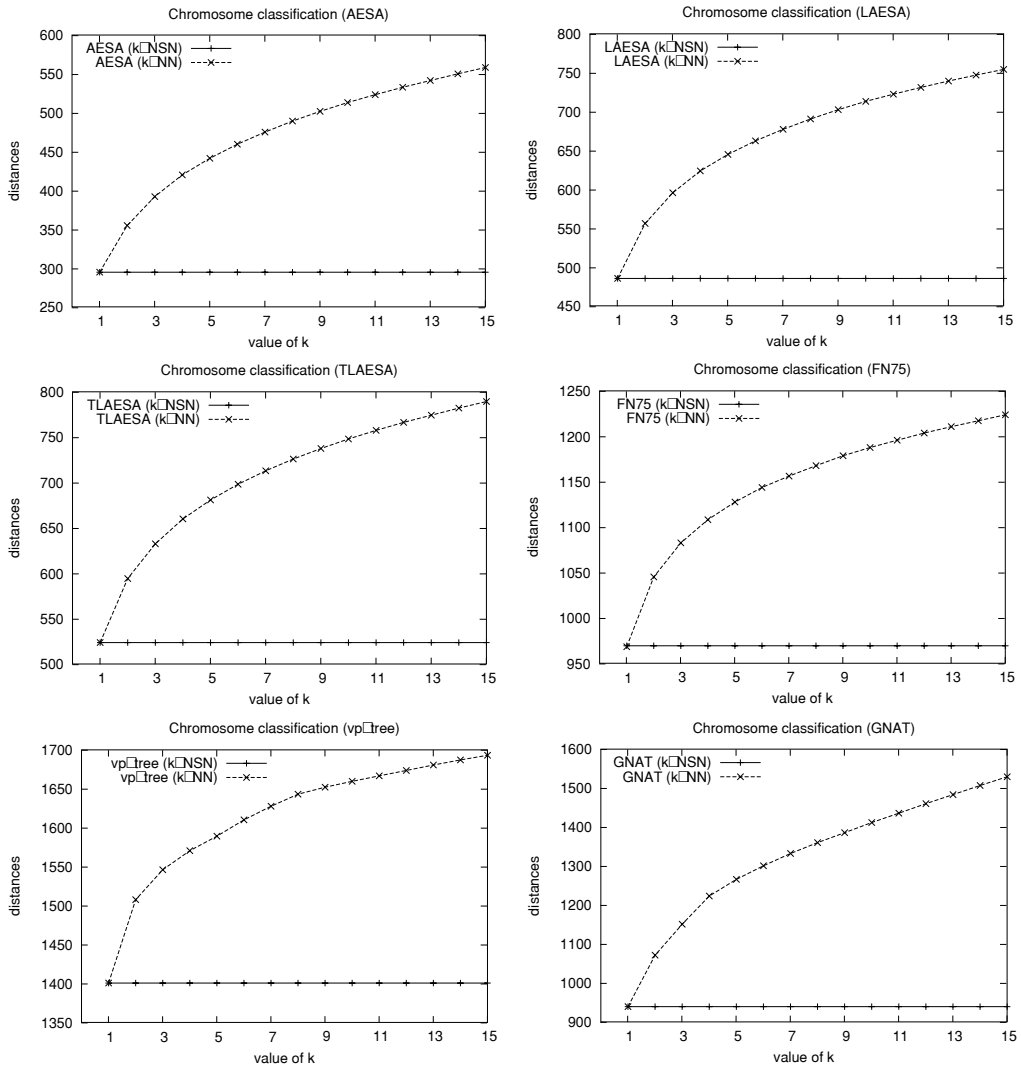


Figure 5: Comparison between k-NN and k-NSN average distance computations in the chromosome classification task.

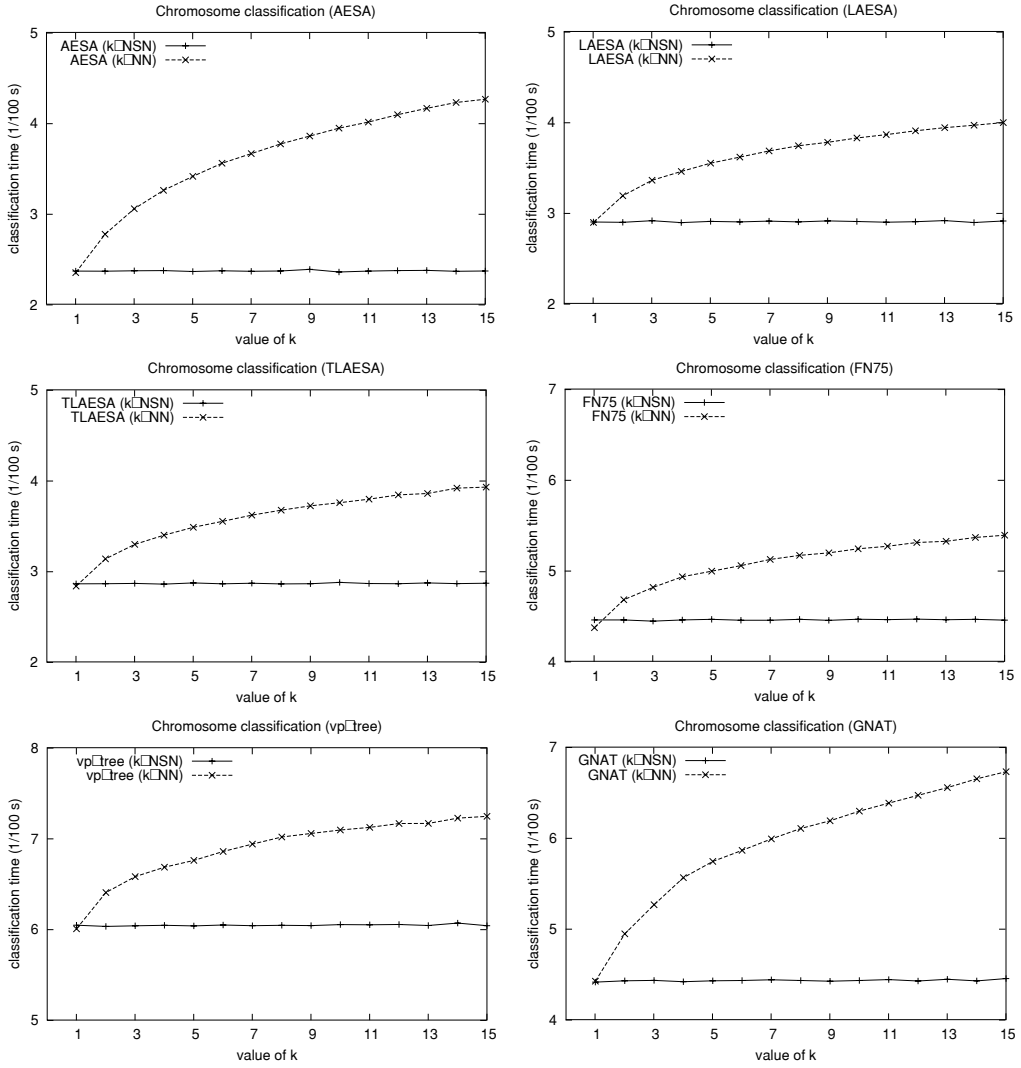
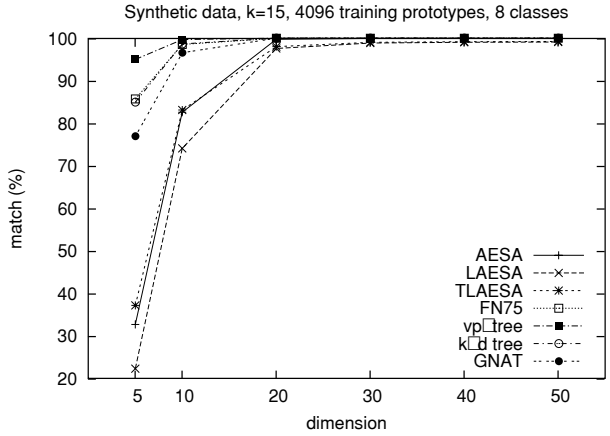
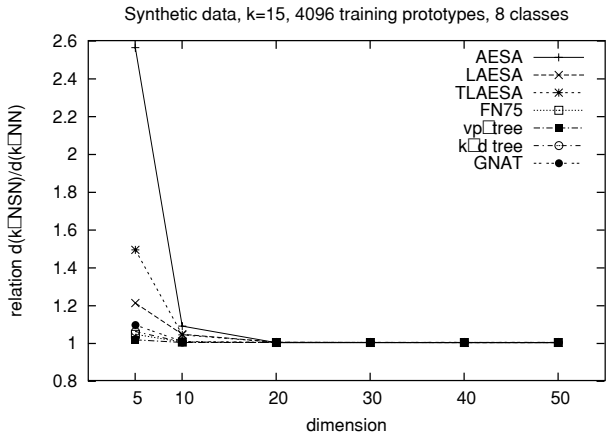


Figure 6: Comparison between k-NN and k-NSN average classification time per sample in the chromosome classification task.



Dim	LAESA	TLAESA	AESA	GNAT	FN75	kd-tree	vp-tree
5	22.26	37.13	32.66	76.91	85.71	84.94	95.00
10	74.09	83.13	82.61	96.55	98.55	98.64	99.69
20	97.61	98.03	99.78	99.98	100	100	100
30	98.91	99.04	100	100	100	100	100
40	99.07	99.19	100	100	100	100	100
50	99.15	99.26	100	100	100	100	100

Figure 7: Percentage of the k-NSN that are among the k-NN, for synthetic data of various dimensions.



Dim	LAESA	TLAESA	AESA	GNAT	FN75	kd-tree	vp-tree
5	1.21	1.49	2.56	1.09	1.04	1.06	1.02
10	1.05	1.04	1.09	1.01	1.00	1.00	1.00
20	1.00	1.00	1.00	1	1	1	1
30	1.00	1.00	1	1	1	1	1
40	1.00	1	1	1	1	1	1
50	1	1	1	1	1	1	1

Figure 8: Relation between the distance to the sample of the kth NSN and the distance of the kth NN, for synthetic data of various dimensions. The value 1.00 indicates that is slightly greater than 1, but not exactly 1.

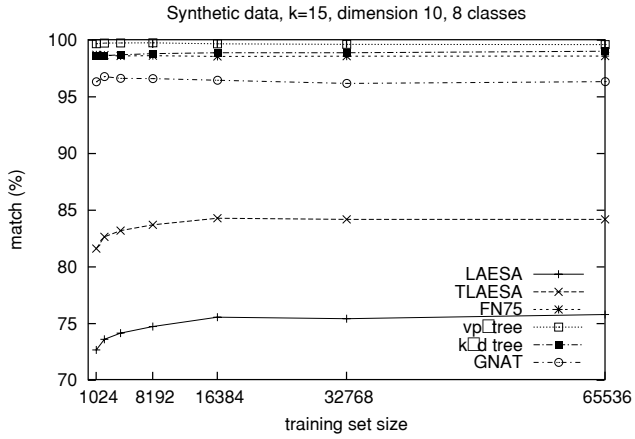


Figure 9: Percentage of the k-NSN that are among the k-NN, for synthetic data with increasing training set size.

5 Experiments with approximate NN search

The goal of approximate NN search is to reduce classification time, maybe slightly increasing error rates. In order to test the approach proposed in section 3, several experiments with both synthetic and real data have been developed. For the synthetic data, the objective was to compare the error rates and distance computations of the algorithms for general metric spaces with the algorithm by Arya et al. [12], in whose ideas we inspired to develop our technique. In the case of real data tasks, we have developed experiments with the chromosome classification task, using the string edit distance.

5.1 Experiments with synthetic data

The implementation the Arya and Mount algorithm was taken from the ANN software package [24], and the experiment was developed with dimension 10 data, using 4096 prototypes for training and 1024 for test. As in previous experiments, 10 different pairs of train/test sets were used. The value of k was set to 25, and in both cases the classification rule was the k-NN rule. Figure 10 plots the error rates and the distances computed by all the algorithms with increasing value for ϵ. The results show that ANN error rates are only beaten by the Fukunaga and Narendra's algorithm, but it computes a higher number of distances; however, the ANN package uses kd-trees, which compute a high number of partial distances (which have

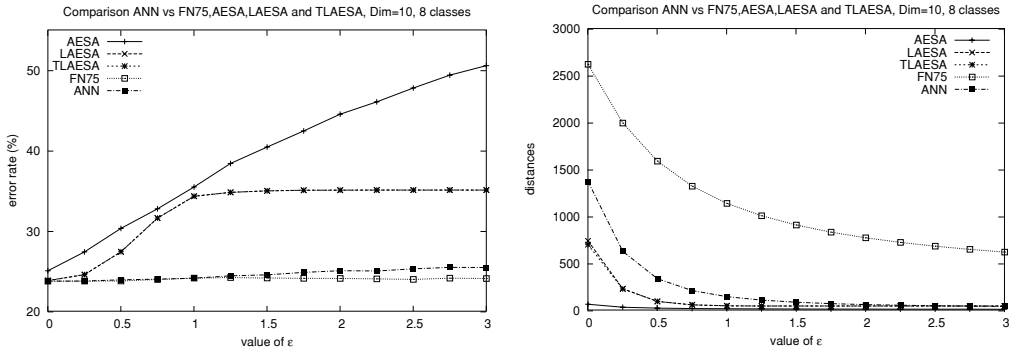


Figure 10: Comparison of error rates (left) and distances computed (right) for the ANN, Fukunaga and Narendra's algorithm, AESA, LAESA and TLAESA.

not been accounted in this experiment). The AESA family algorithms seem to be competitive only for very low values of ϵ, as its error rates increase very quickly with the value of ϵ.

5.2 Experiments with real data

Approximate NN search is specially indicated when the distance is very time consuming, as in the case of the string edit distance in the chromosome classification task mentioned before. Figure 11 plots the results for a simple 1-NN search using various values for ϵ; dotted lines represent error rates, whereas non-dotted lines represent average classification time per sample. As can be appreciated in the figure, the Fukunaga and Narendra's algorithm seems to be the best if we exclude the AESA algorithm, that has a quadratic spatial complexity that limits its applicability. The LAESA and TLAESA results are almost equal due to the fact that both use the same lower bound of the distance from a prototype to the sample.

For the chromosome task the optimum value of k is 11 (see figure 4), and thus we tested the approximate search with $k = 11$ on the Fukunaga and Narendra's algorithm, using both the k-NN and k-NSN rules. The results are plotted in figure 12, and show that using approximate search in combination with the k-NSN rule produces a lower classification time, but with an error rate that is slightly higher than k-NN rate. The important point is that the behaviour of the k-NSN rule remains the same with approximate search than with standard search.

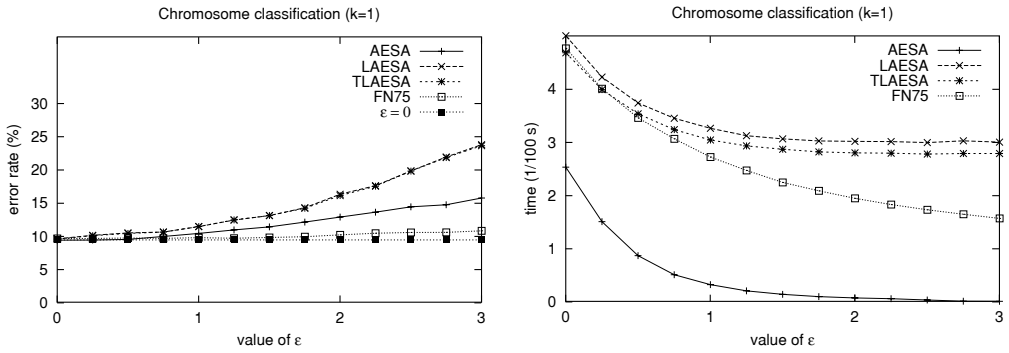


Figure 11: Error rates (left) and classification times per sample (right) in the chromosome classification task, using approximate NN search.

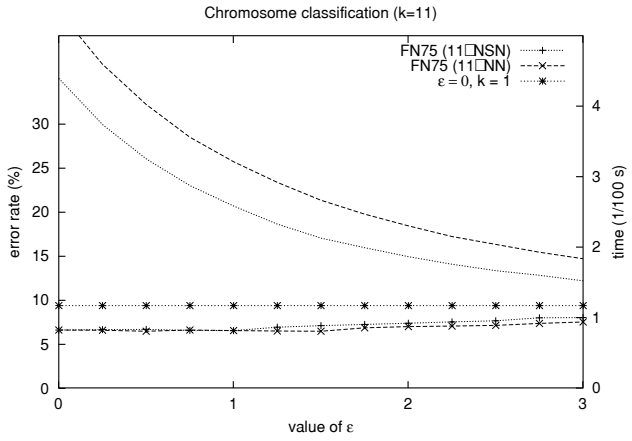


Figure 12: Error rates (lines with points) and classification times per sample (lines without points) in the chromosome task, with the Fukunaga and Narendra's algorithm using both the k-NSN and the k-NN rules, for $k = 11$.

6 Conclusions and future work

A new NN based classification rule (the k-NSN rule) has been developed and tested with various well known fast NN search algorithms, which fit into the approximation and elimination framework: kd-tree, Fukunaga and Narendra's, vp-tree, GNAT. The rule has also been tested with the algorithms of AESA family, which also fit in the approximation and elimination framework.

The experiments show that classification results similar to those of the k-NN rule are obtained using this rule with very little extra computational effort with respect to a NN classifier. Whenever a fast approximation and elimination NN search algorithm is applicable, it may be easily modified to classify using the k-NSN rule and thus it may obtain error rates lower than those of NN, without the extra overhead of searching for the k-NN. Moreover, the time performance of k-NSN classifiers does not depend on the value of k, and the error rates decrease (and get closer to those of the k-NN rule) as the dimensionality increases. The k-NSN rule may be an alternative for the k-NN rule when classification time is an important question in a classification task; also, it may be employed to determine the optimum value for k in the design of a classifier, even if the k-NN rule is finally chosen.

In addition to this rule, previous work on approximate NN search for vector spaces has been extended to algorithms suitable for general metric spaces, and it has been combined with the k-NSN rule, yielding very interesting results for real tasks.

There is still a lot of work to do to explore the possibilities and range of application of the k-NSN rule and approximate NN search. As for the future, we plan to:

- study the evolution of k-NSN error rates as the value of k become higher than those tested in this work, and compare them with k-NN,
- test the performance of the k-NSN rule as the dimensionality or the number of classes increase, and
- apply the k-NSN rule to other approximation-elimination NN search algorithms.
- extend approximate NN search to other algorithms not based on vector spaces, and study its performance in combination with the k-NSN rule.

References

[1] R. Duda and P. Hart. *Pattern Recognition and Scene Analysis*. Wiley, 1973.

[2] K. Fukunaga and M. Narendra. A branch and bound algorithm for computing k–nearest neighbors. *IEEE Transactions on Computing*, 24:750–753, 1975.

[3] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3:209–226, 1977.

[4] P.N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 311–321, 1993.

[5] E. Vidal. New formulation and improvements of the nearest-neighbour approximating and eliminating search algorithm (AESA). *Pattern Recognition Letters*, 15:1–7, 1994.

[6] S. Brin. Near neighbor search in large metric spaces. In *Proceedings of the* 21^{st} *VLDB Conference*, pages 574–584, 1995.

[7] S. Nene and S. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, 1997.

[8] E. Chavez, G. Navarro, R.A. Baeza-Yates, and J.L. Marroquin. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[9] V. Ramasubramanian and K.K. Paliwal. Fast nearest-neighbor search algorithms based on approximation-elimination search. *Pattern Recognition*, 33:1497–1510, 2000.

[10] L. Micó, J. Oncina, and E. Vidal. A new version of the nearest neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing-time and memory requirements. *Pattern Recognition Letters*, 15:9–17, 1994.

[11] F. Moreno-Seco, L. Micó, and J. Oncina. A modification of the LAESA algorithm for approximated k-nn classification. *Pattern Recognition Letters*, 24(1–3):47–53, 2003.

[12] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998.

[13] P. Indyk and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[14] K.L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Computational Geometry*, 22(1):63–93, 1999.

[15] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of 29th ACM Symposium on Theory of Computing*, pages 599–608, 1997.

[16] L. Micó, J. Oncina, and R. C. Carrasco. A fast branch and bound nearest neighbour classifier in metric spaces. *Pattern Recognition Letters*, 17:731–739, 1996.

[17] E. Vidal, F. Casacuberta, and H. Rulot. Is the dtw "distance" really a metric? an algorithm reducing the number of dtw comparisons in isolated word recognition. *Speech Communication*, (4):333–344, 1985.

[18] A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice-Hall, 1988.

[19] C. Lundsteen, J. Phillip, and E. Granum. Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

[20] E. Granum, M.G. Thomason, and J. Gregor. On the use of automatically inferred Markov networks for chromosome analysis. In C. Lundsteen and J. Piper, editors, *Automation of Cytogenetics*, pages 233–251. Springer-Verlag, Berlin, 1989.

[21] E. Granum and M.G. Thomason. Automatically inferred Markov network models for classification of chromosomal band pattern structures. *Cytometry*, 11:26–39, 1990.

[22] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, 1974.

[23] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[24] D.M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching, 1997. url: http://www.cs.umd.edu/ mount/ANN.

Off-line and On-line Continuous Handwritten Text Recognition in PRHLT Group *

Alejandro H. Toselli, Moisés Pastor, Verónica Romero,
Alfons Juan, Enrique Vidal, Francisco Casacuberta

Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia,
46071 Valencia, Spain.
[vromero,ajuan,fcn]@dsic.upv.es
[ahector,moises,evidal]@iti.upv.es

Abstract

The main purpose of this work is to provide a qualitative description of the current research area on Continuous Handwritten Text Recognition for both off-line and on-line cases, carried out by the Pattern Recognition and Human Language Technology (PRHLT) group of the *"Instituto Tecnológico de Informática"*. A general overview of a handwriting recognition system is given, focussing specially on architectonic scheme. According to the case type (off-line and on-line), different preprocessing and feature extraction methods are briefly explained. Also, a short description about how the different linguistic levels: morphological, lexical and syntactical are modelled using the finite-state technology is dedicated. Finally, several fully functional prototype applications of handwriting recognition are presented.

Keywords: Handwritten text recognition, handwriting preprocessing methods, handwriting feature extraction, Hidden Markov Models, handwriting recognition application.

1 Introduction

The off-line and on-line continuous handwritten text recognition are one of the current research areas carried out by the Pattern Recognition and Human Language Technology (PRHLT) group of the *"Instituto Tecnológico de Informática"*. The main purpose here is to provide a qualitative description of the group activity in

* Work supported by the *Agencia Valenciana de Ciencia y Tecnología (AVCiT)" under grant GRUPOS03/031* and the Spanish Project TIRIG (TIC 2003-08496-C)

these areas, starting with a general overview of a handwritten text recognition system focussing specially on architectonic scheme. According to the considered case type (off-line and on-line), different methodologies used for preprocessing and feature extraction are briefly explained, emphasizing on their conceptual basis. Moreover, a short description is offered about how the different linguistic levels: morphological, lexical and syntactical are modelled and integrated together using the finite-state technology.

The recognition problem for both off-line and on-line cases are addressed using *standard* continuous speech technology. Many recent works of the group address this problem in this way (see, among others, [1, 2, 3, 4]).

Here not only recognition has been considered, but also interpretation of the recognized string is required as will be seen for handwritten text applications.

In the next section, a general overview of a handwriting recognition system is shown. Section 3 describes the preprocessing and feature extraction methods more frequently used by the group according to the considered case type. Section 4 is dedicated to the models which the systems are based on. Finally, some functional handwriting recognition prototype applications are presented in the last section.

2 System Overview

The handwritten text recognition system follows the classical architecture. Including both training and recognition phases, it consists basically of five modules:

1. **The preprocessing module**: where line segmentation (just for off-line case only), noise reduction and normalization take place.
2. **The feature extraction module**: where the input of a handwritten text is transformed into a sequence of numerical feature vectors.
3. **The character HMMs models training**: where the HMM parameters are estimated using the Baum-Welch re-estimation algorithm [5].
4. **The lexicon and language models inference**: where they are inferred from the handwritten text image transcriptions. The language model will provide linguistic knowledge about the context in which a word is likely to occur.
5. **The recognition module**: where sequences of feature vectors are converted into word classes.

The fig 1 shows a scheme of a general handwritten text recognition system overview.

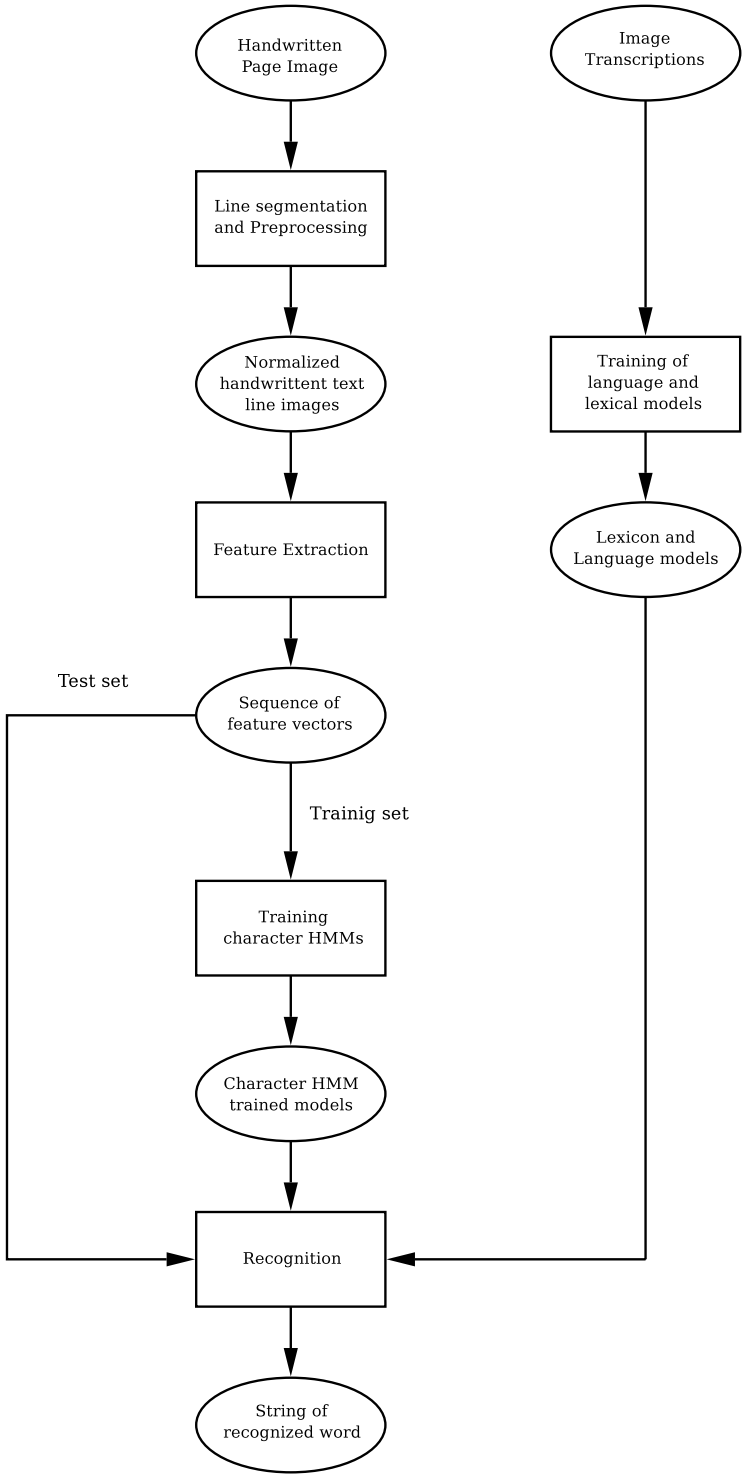


Figure 1: General handwritten text recognition system overview.

3 Preprocessing and Feature Extraction

3.1 Off-Line Case

Preprocessing of handwritten text lines has not yet been given a general, standard solution and it can be said that each handwriting recognition system has its own, particular solution. There are, however, generic preprocessing operations such as *slope* and *slant correction* for which robust techniques are available [2]. But in many cases, other not so generic preprocessing operations are also needed to compensate for a weakness in the ability of the system to model pattern variability. In particular, this is the case of approaches like ours that use (one-dimensional) hidden Markov models for a handwritten text line image. Although these models do properly model (non-linear) horizontal image distortions, they are to some extent limited for vertical distortion modeling. Therefore, apart from the usual slope and slant correction preprocessing steps, it has been included a third step aimed at reducing a major source of vertical variability: the height of ascenders and descenders. These steps are discussed hereafter.

Slope correction module processes an original image to put the text line into horizontal position. As each word or multi-word segment in the text line may be skewed at a different angle, the original image is divided into segments surrounded by wide blank spaces and slope correction is applied to each segment separately. This is not to obtain a segmentation of the text line into words and it is not necessary for each segment to contain exactly one word. In the fig. 2 is illustrated one of the used slope correction methods which is carried out in four steps: a) horizontal run-length smoothing of the segments comprising the original image (panel b.1 in fig. 2); b) computation of the upper and lower contours for each segment (panel b.2); c) eigenvector line fitting of the contours (panels b.3 and b.4); and d) segment deskewing in accordance to the average angle of the contour lines (panel b.5).

Slant correction shears the deskewed image horizontally to bring the writing in an upright position. Following the procedure described in [4], the dominant slant angle of the writing is obtained based on projection profile.

As said above, the third step is aimed at reducing a major source vertical variability: the height of ascenders and descenders (not that of the main text body). The reference lines computed for each image segment during slope correction are updated and joined together to separate the main text body from the zones with ascenders and descenders. Then, each of these zones is linearly scaled in height to a size determined as a percentage of the main body vertical size. Since these zones are often large, nearly blank areas, this scaling operation has the effect of filtering out most of the uninformative background. It also compensates for the large variability

of the ascenders and descenders height as compared with that of the main text body.

As with any approach based on (one-dimensional) hidden Markov models, feature extraction is required to transform the preprocessed image into a *sequence of (fixed-dimension) feature vectors*. To do this, the preprocessed image is first divided into a grid of square cells whose size is a small fraction of the image height (such as 1/16, 1/20, 1/24 or 1/28). We call this fraction *vertical resolution*. Then each cell is characterized by the following features: *normalized grey level*, *horizontal grey-level derivative* and *vertical grey-level derivative*. To obtain smoothed values of these features, feature extraction is extended to a 5×5 window centered at the current cell weighted with a Gaussian function. The derivatives are computed by least squares fitting of a linear function.

Columns of cells are processed from left to right and a feature vector is built for each column by stacking the features computed in its constituent cells (panel e in fig. 2). This process is similar to that followed by Bazzi *et al* [6].

3.2 On-Line Case

The main characteristic which determine the on-line nature is its input. The on-line input data stream consists of a sequence of strokes. A stroke consists on a sequence of coordinates ordered in time (x_t, y_t), that is a curve. There are two kinds of strokes: pen-down strokes (also referred to as visible strokes) acquired with the digital pen touching the pad surface, and pen-up strokes acquired without touching it. Because of the visible-stroke information is effectively found in the pen-down strokes, pen-up strokes are not considerated.

The preprocessing of each sample involves six processes: repeated points elimination, noise reduction, slope and slant normalization, size normalization and writing speed normalization. Noise here, has different nature compared with off-line images. The text is introduced directely without any kind of intermediate support, thus the background does not exists. Noise in handwritten strokes is due to erratic hand motions and inaccuracy of the digitalization process. In order to reduce noise, we employ a smoothing technique consisting in replacing every point (x_t, y_t) in the trajectory by the mean value of its neighbors [7]. It is important to remark that the temporal order of the data points is preserved throughout all preprocessing steps.

To correct the slope, the local minima point must be found for all strokes. Anomalous points are eliminated. A line is adjusted using the eigenvector method. Once, the line angle is determined, the image is corrected with a rotation operation.

For each line between two consecutive points, the angle is computed, then the histogram of these angles is built. The slant angle is calculated by searching for the

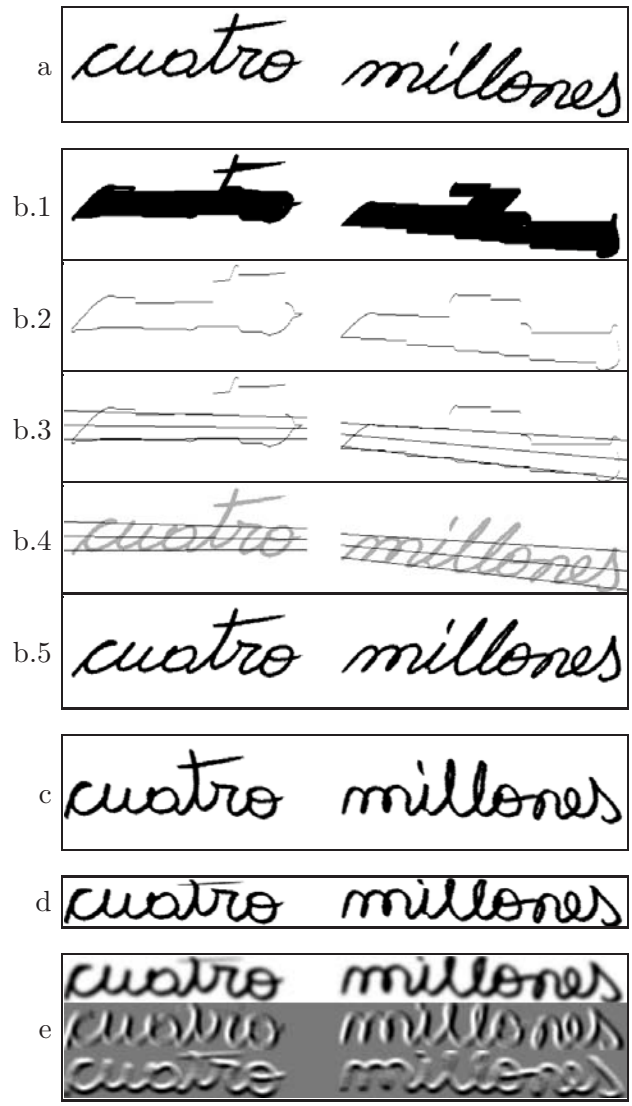


Figure 2: Preprocessing and feature extraction example. From top to bottom: a) original image ("four millions" in Spanish); b) skew angle estimation and correction (block of 5 joint panels); c) slant correction; d) height normalization for ascenders and descenders; and e) extracted sequence of feature vectors (normalized grey levels, horizontal derivatives and vertical derivatives). From top to bottom in the block of 5 joint panels describing skew angle estimation and correction: b.1) horizontal run-length smoothing of the two segments (words) comprising the original image; b.2) upper and lower contours; b.3) eigenvector line fitting of the contours; b.4) fitted lines; and b.5) deskewed image.

most frequent value in the angle histogram. To correct the slant, a shear operation must be done.

Other commonly applied on-line HTR preprocessing operation is the so-called *trace segmentation*, which consists in a resampling operation that redistribute data points (originally sampled in equal time intervals) to enforce even spacing between them. This way the word will have the same points at the same places independently of the speed the word was written. Trace segmentation is used not only for speed invariance, but also to reduce the size of the samples and speed up the recognition time. In [8], PRHLT group carried out a study about the trace segmentation resampling distance effect (figure 3 shows some processed word examples for different resampling distances). As an alternative to trace segmentation is the use of normalized derivatives. Derivatives explain both the direction and the speed of the trace. If the module value of transformed derivatives becomes constant (equal to 1), the representation will be invariant to the writing speed, while keeping the direction information. It is shown that the use of *"normalized derivatives"* leads to better results compared with trace segmentation methodology.

Once the original coordinate sequences have been preprocessed they are transformed into new temporal sequences of 6-dimensional real-valued feature vectors. The six features computed for each sample point are: normalized vertical position (y_{N_t}, within the range $[0, 100]$), first derivatives (x', y') calculated using the method given in [9], second derivatives (x'', y'') computed in the same way as the first derivatives and curvature (k_t) which is the inverse of the radius of the curve in each point. It is worth noting that the discarded pen-up strokes still remain implicit in the transition from each last point of a pen-down stroke and the initial point of its following pen-down stroke. These transitions are characterized by first derivatives huge values.

The coordinate x is not used as a feature because of x range for different instances of the same character, can vary greatly depending on the position of the character into a word.

4 Modelling Scheme

Sentence models are built by concatenation of *word* models which, in turn, are often obtained by concatenation of continuous left-to-right HMMs for individual *characters*.

Basically, each character HMM is a stochastic finite-state device that models the succession, along the horizontal axis for off-line case and time for on-line, of feature vectors extracted from instances of this character. Each HMM state generates feature vectors following an adequate parametric probabilistic law; typically, a *mixture*

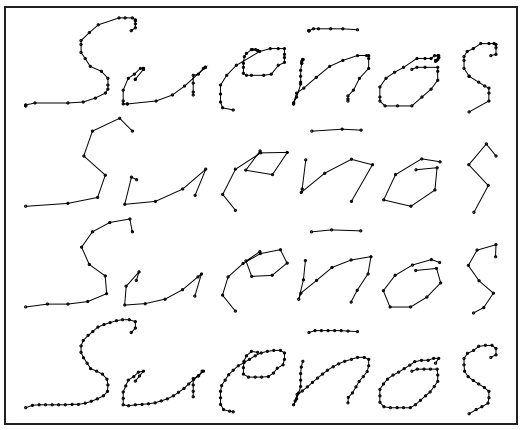


Figure 3: From top to bottom: original image, trace segmentation with resampling distance $\alpha = 60$, with $\alpha = 40$ and with $\alpha = 13$ (the best performer). Note: in the case of *Derivative Normalization* approximation, points remain at their position, the image produced is similar to the original one.

of Gaussian densities. The adequate number of densities in the mixture per state, as well as the number of HMM states, need to be tuned empirically and it may be conditioned by the available amount of training data.

Once an HMM "*topology*" (number of states and structure) has been adopted, the model parameters can be easily trained from images of continuously handwritten text (*without any kind of segmentation*) accompanied by the transcription of these images into the corresponding sequence of characters. This training process is carried out using a well known instance of the EM algorithm called *forward-backward or Baum-Welch re-estimation* [5].

From this point on, it is worth remarking that does not exist any difference in the modelling scheme used by off-line and on-line cases, so the description hereafter is the same for both.

Words are obviously formed by concatenation of characters. In our finite-state modeling framework, for each word, a stochastic finite-state automaton (SFS) is used to represent the possible concatenations of individual characters to compose this word. This automaton takes into account possible inter-word blank spaces, as well as optional character capitalizations. Fig. 4 shows an example of character HMM (left) and SFS automaton word (right).

Sentences are formed by the concatenation of words. This concatenation is modeled by SFS model automatically learned from training data [10] or built by hand in accordance with previous knowledge about the task. Usually this SFS are

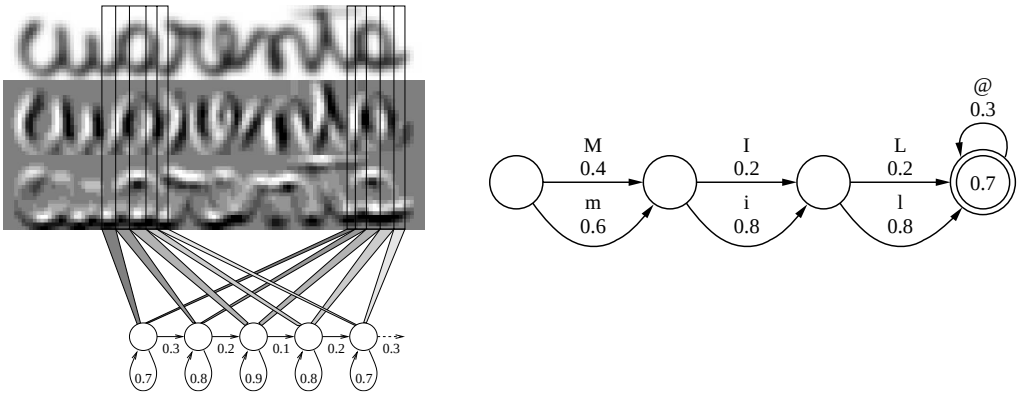


Figure 4: An example of character HMM for off-line case and automaton word. HMM modeling of instances of the character "a" within the Spanish word "cuarenta" (forty). The states are shared among all the instances of characters of the same class. Automaton for the lexicon entry "mil" (One hundred). The symbol "@" represents a blank segment.

n-grams [5], which uses the previous $n-1$ words to predict the next one and can be max-likelihood learned from a training (text) corpus, by simply counting relative frequencies of n-word sequences in the corpus [5].

4.1 Recognition via Finite-State Models

Due to the *homogeneous* nature of all these finite-state (character, word and sentence) models, they can be easily *integrated* into a single *global* SFS model that accepts sequences of raw feature vectors and outputs strings of recognized words. To this end each edge of the SFS sentence is expanded by a *concatenation* of the HMMs of the successive characters which constitute the source-language word of this edge. To deal with possible inter-word white space a *blank* ("@") special HMM can be trained and also integrated in the network. This network expansion, illustrated in Fig. 5, realizes the integration of *character, word and sentence* levels.

Given an input sequence of feature vectors, the best output hypothesis is one which corresponds to a series of states of the integrated model that, with highest probability, produces the input feature-vector sequence. This global search process is very efficiently carried out by the well known (*beam-search*-accelerated) Viterbi algorithm [5]. This technique allows integration to be performed "on the fly" during the decoding process. In this way, only the memory strictly required for the search is actually allocated.

5 Off-Line and On-Line HTR Applications

A set of implemented applications based on off and on-line HTR technology are presented hereafter. It is worth noting that not only recognition has been considered, but also interpretation of the recognized string. This is so especially for the two following applications.

5.1 Handwriting Recognition System for Spanish Numbers

This is a syntax-constrained *interpretation* application, where the *recognition* consists in getting adequate hypotheses about the handwritten words (see figure 6), while the goal of *interpretation* is to come out with a *numeric* expression which, overall, reflects what was written in letters as accurately as possible. It is not of great importance whether all the words comprising the legal amount were correctly written or whether they can be exactly recognized or not; only the reliability of the *interpreted* numeric result really matters.

The interpretation is feasible, due to the use of a hand-built (*sequential*) stochastic finite-state transducer [10] (SFST), that accepts any text Spanish number in the range from 0 to $10^{12} - 1$ and outputs an *arithmetic expression* giving its corresponding numerical value. A small fragment of this transducer is shown in Fig. 7.

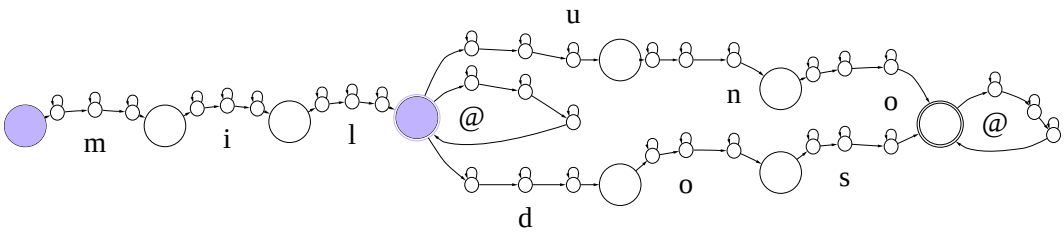


Figure 5: A small piece of an integrated finite-state model, using three-state character HMMs. The part shown stands for the sentences “mil”, “mil uno” and “mil dos” (*1,000; 1,001; 1,002*).

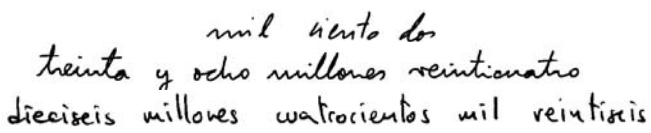


Figure 6: Examples of real continuous text sentences of Spanish numeric amounts: *1,102*; *38,000,024*; *16,400,026*.

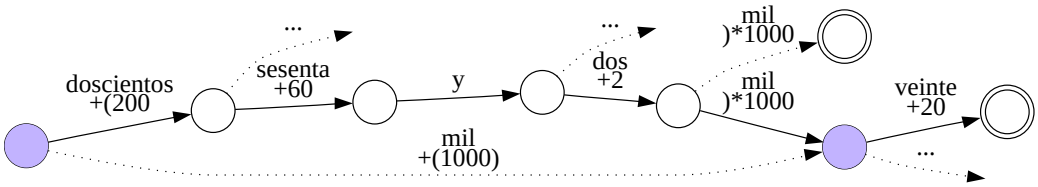


Figure 7: A piece of the hand-designed *numbers transducer*. Solid-line edges correspond to a path that accepts *"doscientos sesenta y dos mil veinte"* (two hundred sixty two thousand and twenty), yielding "*+(200+60+2)*1000+20*".

In other words, its output is an *arithmetic expression* whose value is that of the number given through the input text; for example, from the Spanish text *"doscientos sesenta y dos mil veinte"* (two hundred sixty two thousand and twenty) the provided output is: "$+(200+60+2)*1000+20$". From this expression the target (decimal) number (262,020) can be easily obtained easily from a simple postprocessing, piping the output of the SFST to the standard Unix tool "`bc`".

The figure 8 illustrates the full interpretation process carried out starting with the usual preprocessing stage (already described in figure 2), following this, the feature extraction phase and finishing with the output recognition-interpretation *arithmetic expression* hypothesis.

This application resembling legal amount interpretation for bank checks, where a reading system has to *interpret* the legal amount (written in letters) to determine the real numeric sum (and to optionally verify whether this sum matches the courtesy amount – written in digits).

5.2 Classification of Spontaneous Handwriting Answers

Here a handwritten text recognition and classification application entailing casual, spontaneous writing and a relatively large vocabulary is considered. In this application, however, the extreme difficulty of text recognition is somehow compensated by the simplicity of the target result. The application consists of classifying (into a small number of predefined classes) casual handwritten answers extracted from survey forms made for a telecommunication company.[1]

The considered application phrases were handwritten by a heterogeneous group of people, without any explicit or formal restriction relative to vocabulary, the resulting application lexicon becomes quite large. On the other hand, since no guidelines are given as to the kind of pen or the writing style to be used, phrase become very

[1]Data kindly provided by ODEC, S.A. (`www.odec.es`)

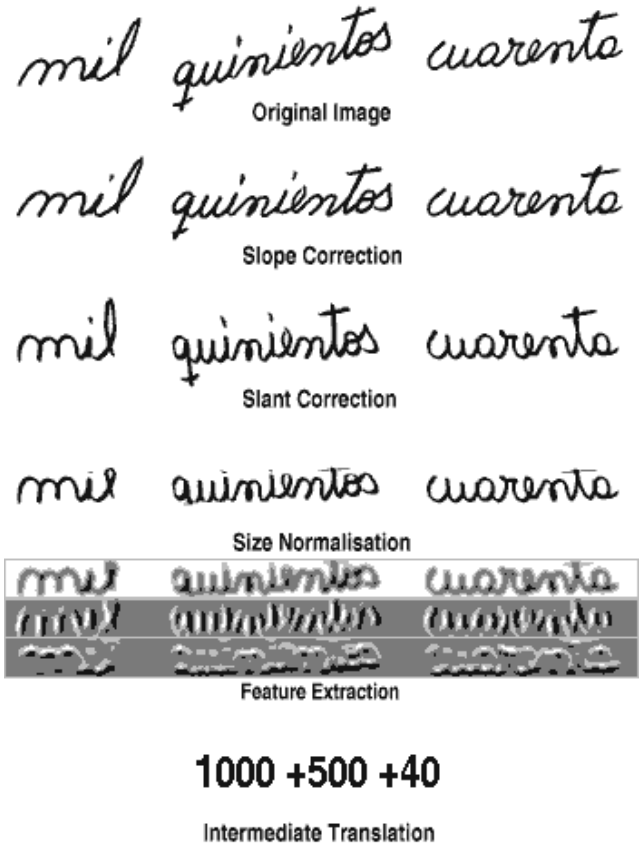


Figure 8: Example of preprocessing, feature extraction and output recognition-interpretation *arithmetic expression* hypothesis of the Spanish text *"mil quinientos cuarenta"* (five hundred forty).

variable and noisy. For example, in some samples the stroke thickness is non-uniform and the vertical slant also varies within a sample. Other samples present irregular and non-consistent spacing between words and characters. Also, there are samples written using different case and font types, variable sizes and even including foreign language phrases. On the other hand, noise and non-textual artifacts often appear in the phrases. Among these noisy elements we can find unknown words or words containing orthographic mistakes, as well as underlined and crossed-out words. Unusual abbreviations and symbols, arrows, etc. are also within this category. The combination of these writing-styles and noise may result in partly or entirely illegible samples. Examples of these difficulties are shown in Figure 9. So far, human

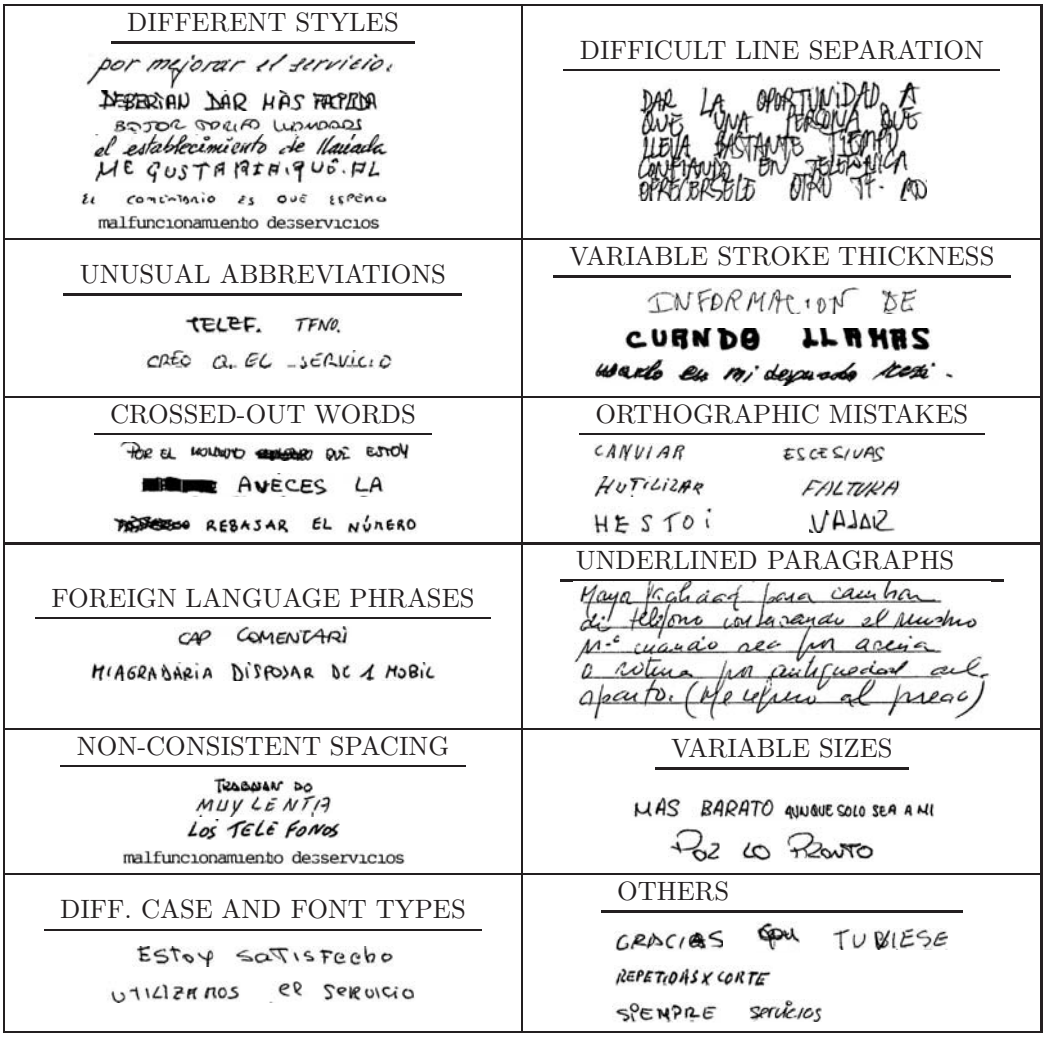


Figure 9: Some of the difficulties involved in the application.

operators have been in charge of classifying these phrases. They do it through a fast reading which just aims to grasp the essential meaning of the answers. This implies that not all the words can or need to be perfectly recognized; they just retrieve enough information to get an adequate classification. In particular, the eight classes defined in the application are: telephone rates, coverage problems, mobile telephone problems, customer assistance, customers expressing satisfaction, service complains and generic queries for information. The aim of our system is to help performing this classification as fast and accurately as possible, with a minimal human intervention.

In [1, 11], the group has proposed to tackle this difficult classification task using a *two-step* or *serial approach*. Using character HMMs integrated with an n-gram language model, recognition is first performed on each handwritten sample; then, the recognized word sequence is classified into one of the given eight classes using a text classifier based on either n-grams or multinomials. Figure 10 shows three examples of handwritten phrase images along with their recognition-classification results.

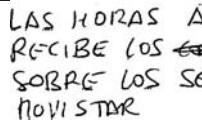
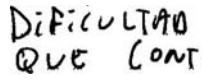
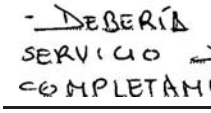
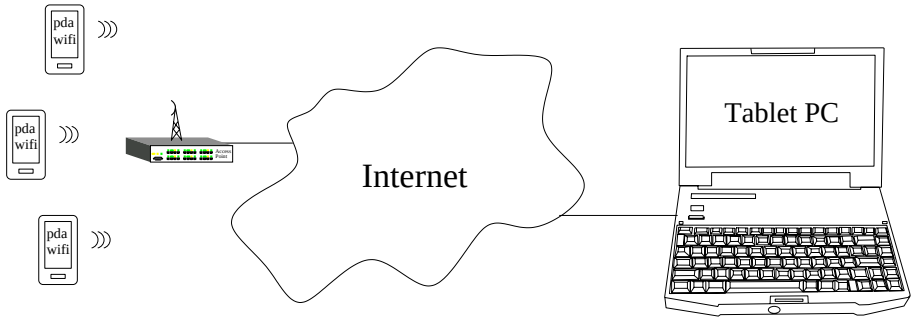
Image	Recognition Result	Classif. Result
	**BIEN OTRAS D** LAS QUE SE RECIBE LOS **CORREO** MENSAJES SOBRE LOS SERVICIOS **A P** MOVISTAR	Wrong
	DIFICULTAD EN SABER QUE CONTRATO **CAMBIAR**	Correct
	DEBERÍA TENER UN SERVICIO **EN E** NOTICIAS COMPLETAMENTE GRATIS	Correct

Figure 10: Examples of three handwritten phrases along with their recognition and classification results. The misrecognized words are indicated in underlined bold-face.

5.3 Handwriting recognition application for Tablet PC

A Tablet PC is a relatively new generation of portable PC that has a touch screen and whose main method of input is handwriting recognition. The group has been developed an on-line handwriting recognition prototype application to run on one of this machines. This application can recognize from isolated handwritten characters to whole sentences.

The sequence of coordinates (x_t, y_t) are provided directely by the tablet pc input panel, but also can be received from any device connected to Internet, as for example, *personal digital assistants* (PDA). The figure shows this connection scheme.



This application is intended to be used in hospitals where medical personal could employ PDA's as a mean to take handwritten notes about, for example, clinical diagnostic of patients. These notes are send to a central computer where they are recognized and stored in a data base.

References

[1] Alejandro H. Toselli, Alfons Juan, and Enrique Vidal. Spontaneous Handwriting Recognition and Classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 433–436, Cambridge, United Kingdom, August 2004.

[2] A. H. Toselli, A. Juan, D. Keysers, J. González, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta. Integrated Handwriting Recognition and Interpretation using Finite-State Models. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 18(4):519–539, June 2004.

[3] J. González, I. Salvador, A. H. Toselli, A. Juan, E. Vidal, and F. Casacuberta. Off-line Recognition of Syntax-Constrained Cursive Handwritten Text. In *Proc. of the S+SSPR 2000*, pages 143–153, Alicante (Spain), 2000.

[4] Moisés Pastor, Alejandro Toselli, and Enrique Vidal. Projection profile based algorithm for slant removal. In *International Conference on Image Analysis and Recognition (ICIAR'04)*, Lecture Notes in Computer Science, pages 183–190, Porto, Portugal, September 2004. Springer-Verlag.

[5] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[6] I. Bazzi, R. Schwartz, and J. Makhoul. An Omnifont Open-Vocabulary OCR System for English and Arabic. *IEEE Trans. on PAMI*, 21(6):495–504, 1999.

[7] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. On-Line Handwriting Recognition: The NPen++ Recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–181, 2001.

[8] M.Pastor, A.H.Toselli, and E.Vidal. Writing speed normalization for on-line handwritten text recognition. In *Eighth International Conference on Document Analysis and Recognition (ICDAR05)*, volume II of *Lecture Notes in Computer Science*, pages 1131–1135. Seul (Korea), August 2005.

[9] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Wo odland. *The HTK Book: Hidden Markov Models Toolkit V2.1.* Cambridge Research Laboratory Ltd, March 1997.

[10] J. Oncina, P. García, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15(5):448–458, May 1993.

[11] Alejandro Héctor Toselli, Moisés Pastor, Alfons Juan, and Enrique Vidal. Spontaneous Handwriting Text Recognition and Classification using Finite-State Models. In *Iberian Conference on Pattern Recognition and Image Analysis*, volume 3523 of *Lecture Notes in Computer Science*, pages 363–370. Springer-Verlag, Estoril (Portugal), June 2005.

The naive Bayes model, generalisations and applications*

Vicent Alabau, Jesús Andrés, Francisco Casacuberta, Jorge Civera
José García-Hernández, Adrià Giménez, Alfons Juan,
Alberto Sanchis, Enrique Vidal

DSIC/ITI, Universitat Politècnica de València.
46022 València (Spain)

Abstract

The naive Bayes classification model is a very simple classification technique in which pattern features are assumed to be class-conditional independent. This is the so-called naive Bayes or independence assumption. In spite of being a strong, unrealistic assumption, the naive Bayes model often provides good results at low cost in terms of model complexity. The Pattern Recognition and Human Language Technology group from the Universitat Politècnica de València maintains an active research line on this model, its generalisations (mainly discrete mixture models) and applications (text classification, word disambiguation and confidence measures for speech recognition, etc.).

Keywords: naive Bayes, mixtures, Bernoulli, multinomial, classification

1 Introduction

One of the simplest and most popular classification models is the naive Bayes classifier. Its simplicity its due to the so-called naive Bayes assumption: features are assumed to be independent given the class. In spite of being a strong, unrealistic assumption, this classifier often provides good results. It is widely used for discrete data; e.g. text data. However, the naive Bayes model has been recently outperformed by techniques such as boosting-based classifier committees and support vector machines. Nevertheless, the performance of the naive Bayes classifier can be significantly improved by using generalisations such as finite mixtures [1, 2, 3] or other recent generalisations (and corrections) [4, 5, 6, 7, 8].

*This work was partially supported by the Spanish "Ministerio de Ciencia y Tecnología" under grant DPI2001-0880-CO2-02, the EU project "TT2" (IST-2001-32091), the "Agencia Valenciana de Ciencia y Tecnología" under grant GRUPOS03/031, under grant FPI(CTBPRA/2005/004) and by the Spanish project ITEFTE(TIC2003-08681-C02-02).

The Pattern Recognition and Human Language Technology group from the Universitat Politècnica de València maintains an active research line on this model, its generalisations and applications. Most of the generalisations proposed are based on the use of finite mixtures. These models have been applied to text classification, OCR and other tasks with good results.

The structure of this document is as follows. In Section 2, the naive Bayes model and generalisations are described; this includes the Bernoulli and multinomial instantiations, and its corresponding mixture extensions. Also, proposals for length modelling and bilingual data are described. In Section 3, the results of applying naive Bayes models to different tasks, on which we are currently working, are presented. These tasks are OCR, estimation of confidence measures for speech recognition, word disambiguation and text classification. Finally, some concluding remarks are given in Section 4.

2 The naive Bayes model and generalisations

The Bayes classifier decides that the class of a given sample x is the class $c(x)$ of maximum posterior probability:

$$c(x) = \underset{c}{\operatorname{argmax}}\ p(c \mid x) \tag{1}$$

This classifier gets his name from the Bayes rule:

$$\begin{aligned} p(c \mid x) &= \frac{p(x,c)}{p(x)} \\ &= \frac{p(x,c)}{\sum_{c'} p(x,c')} \\ &= \frac{p(c)\,p(x \mid c)}{\sum_{c'} p(c')\,p(x \mid c')} \end{aligned} \tag{2}$$

where $p(c)$ is the prior probability of class c, and $p(x \mid c)$ is the class c-conditional probability of x. Since the denominator does not depend on c, the classification problem can be expressed as:

$$\begin{aligned} c(x) &= \underset{c}{\operatorname{argmax}}\ \frac{p(c)\,p(x \mid c)}{\sum_{c'} p(c')\,p(x \mid c')} \\ &= \underset{c}{\operatorname{argmax}}\ p(c)\,p(x \mid c) \end{aligned} \tag{3}$$

The naive Bayes is a particular case in which features are assumed to be class-conditional independent. Formally, if $\mathbf{x}$ is a D-dimensional vector, we have:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \ \frac{1}{Z(\mathbf{x})} p(c) \prod_{i=1}^{I} p(x_i \mid c) \tag{4}$$

where $Z(\mathbf{x})$ is a scaling factor dependent only on $\mathbf{x}$.

2.1 Bernoulli instantiation

Let $\mathbf{x}$ be a D-dimensional bit vector. A conventional naive Bayes classifier for binary data is based on the multidimensional Bernoulli distribution:

$$p(\mathbf{x}) = \prod_{d} p(x_d) \tag{5}$$

with

$$p(x_d) = p_d^{x_d} \left(1 - p_d\right)^{1 - x_d} \tag{6}$$

where, for all $d = 1, \ldots, D$, p_d is the probability for bit d of being one. Note that the probability of each bit is independent of other bit values.

The Bernoulli classifier equals the Bayes classifier in the particular case of class-conditional Bernoulli distributions. Thus, we have:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \ p(c) \prod_{d} {p_{cd}}^{x_d} \left(1 - p_{cd}\right)^{1 - x_d} \tag{7}$$

2.2 Multinomial instantiation

Another particular case of the Bayes classifier which is also a naive Bayes model is the multinomial classifier. Let $\mathbf{x}$ be a D-dimensional vector of non-negative integer counts summing up to a given positive integer constant x_+. The multinomial distribution has the following probability function:

$$p(\mathbf{x}) = \frac{x_+!}{\prod_d p(x_d)} \tag{8}$$

with

$$p(x_d) = p_d^{x_d} \tag{9}$$

where, for all $d = 1, \ldots, D$, p_d is the probability of the event whose number of occurrences is given by x_d. As is in the Bernoulli instantiation, the value of x_d does not depend on the values of other features.

In the case of class-conditional multinomial distributions, the Bayes classifier is:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\ p(c)\,\frac{x_+!}{\prod_d x_d!}\prod_d {p_{cd}}^{x_d} \tag{10}$$

2.3 Length modelling

Length modelling for the multinomial text classifier is a well-known problem which is often disregarded. To tackle this problem, we have studied the following estimation technique for the class-conditional probability of a length l:

$$\hat{p}(l \mid c) = \frac{N(c,l)}{\sum_{l'=1}^{\infty} N(c,l')} \tag{11}$$

where $N(c,l)$ is the number of documents of class c having length l.

Another possibility consists of modelling the length as a continuous density function and compute the length probability by integration [9]. The distribution proposed is the Gamma distribution, since it has an infinity tail and its shape is very similar to the table of frequencies of text sentence lengths. Formally, we have:

$$\begin{aligned}\hat{p}(l \mid c) &= \int_{l-1/2}^{l+1/2} Gamma(t \mid \alpha_c, \beta_c)\,\partial t \\ &= \int_{l-1/2}^{l+1/2} \frac{{\beta_c}^{\alpha_c}}{\Gamma(\alpha_c)}\, t^{\alpha_c - 1} e^{-\beta_c t}\,\partial t \\ &\approx \frac{{\beta_c}^{\alpha_c}}{\Gamma(\alpha_c)}\, l^{\alpha_c - 1} e^{-\beta_c l}\end{aligned} \tag{12}$$

For the case of bimodal length distributions, we have also studied the use of two-component finite mixture models. An example is shown in Figure 1.

2.4 Finite mixture models

A mixture of C components is a probability (density) function of the form:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{c=1}^{C} f(\mathbf{x}, c) \\ &= \sum_{c=1}^{C} f(c)\, f(\mathbf{x} \mid c) \\ &= \sum_{c=1}^{C} p_c\, f(\mathbf{x} \mid c, \boldsymbol{\Theta}') \end{aligned} \tag{13}$$

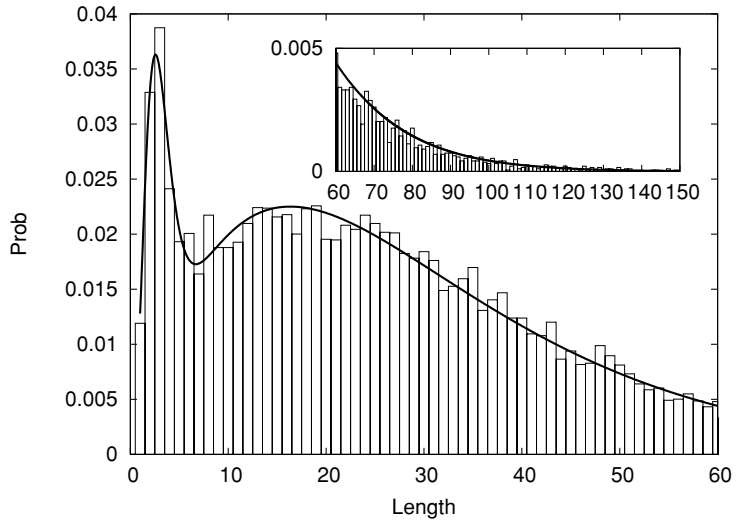


Figure 1: Example of a bimodal length distribution modelled by a two-component mixture of Gamma distributions.

where we assume that $p_c = f(c)$ and $f(\mathbf{x} \mid c) = f(\mathbf{x} \mid c, \mathbf{\Theta}')$. Using this assumption, the unknown model parameters are:

$$\mathbf{\Theta} = \begin{pmatrix} \mathbf{p} \\ \mathbf{\Theta}' \end{pmatrix} \quad with \quad \mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_C \end{pmatrix} \tag{14}$$

In the most simple case we assume that the parameters of each component are independent of the parameters of the rest of components. Thus, we have:

$$\mathbf{\Theta}' = (\mathbf{\Theta_1}', \mathbf{\Theta_2}', \ldots, \mathbf{\Theta_3}') \tag{15}$$

and

$$f(\mathbf{x}) = \sum_{c=1}^{C} p_c \, f(\mathbf{x} \mid c, \mathbf{\Theta_c}') \tag{16}$$

To train a finite mixture model we use the EM algorithm assuming that $\mathbf{x_n}$ is a incomplete sample in the sense that it has lost the label of the component from

which it has been generated [10]. This label can be written as an indicator vector:

$$\mathbf{Z} = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_C \end{pmatrix} \in \{0,1\}^C \tag{17}$$

A value of one in z_i indicates that the sample has been generated by the ith component. Formally, $\mathbf{Z}$ is a multinomial variable of length one,

$$\mathbf{Z} \sim Mult_C(1, \mathbf{p}) \tag{18}$$

with the following probability function:

$$f(\mathbf{z} \mid \mathbf{p}) = \prod_c {p_c}^{z_c} \tag{19}$$

The so-called complete model is:

$$\begin{aligned} f(\mathbf{x}, \mathbf{z}) &= f(\mathbf{z} \mid \mathbf{p})\, f(\mathbf{x} \mid \mathbf{z}, {\boldsymbol{\Theta}_\mathbf{c}}') \\ &= \prod_c \left(p_c\, f(\mathbf{x} \mid z_c = 1, {\boldsymbol{\Theta}_\mathbf{c}}') \right)^{z_c} \end{aligned} \tag{20}$$

from which the original, incomplete model (13) can be obtained by simple marginalisation:

$$\begin{aligned} f(\mathbf{x}) &= \sum_\mathbf{z} f(\mathbf{x}, \mathbf{z}) \\ &= \sum_\mathbf{z} \prod_c \left(p_c\, f(\mathbf{x} \mid z_c, {\boldsymbol{\Theta}_\mathbf{c}}') \right)^{z_c} \\ &= \sum_c p_c\, f(\mathbf{x} \mid z_c = 1, {\boldsymbol{\Theta}_\mathbf{c}}') \end{aligned} \tag{21}$$

In the E step, the function $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)})$ is defined and the expected value of unknown variables is calculated from the value of parameters in iteration k, $\boldsymbol{\Theta}^{(k)}$. So the function $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)})$ is:

$$\begin{aligned} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) &= \sum_n E(\, \log f(\mathbf{x_n}, \mathbf{z_n} \mid \boldsymbol{\Theta}) \mid \mathbf{x_n}, \boldsymbol{\Theta}^{(k)}\,) \\ &= \sum_{n,c} {z_{nc}}^{(k)} \left(\log p_c + log f(\mathbf{x_n} \mid z_{nc} = 1, {\boldsymbol{\Theta}_\mathbf{c}}') \right) \end{aligned} \tag{22}$$

where ${z_{nc}}^{(k)}$ is calculated as:

$$\begin{aligned} {z_{nc}}^{(k)} &= E(z_{nc} \mid \mathbf{x}, \mathbf{\Theta}^{(k)}) \\ &= \frac{{p_c}^{(k)} f(\mathbf{x_n} \mid z_{nc} = 1, {\mathbf{\Theta_c}'}^{(k)})}{\sum_{c'} {p_{c'}}^{(k)} f(\mathbf{x_n} \mid z_{nc'} = 1, {\mathbf{\Theta_{c'}}'}^{(k)})} \end{aligned} \tag{23}$$

In the M step the value of $\mathbf{\Theta}$ that maximises the expression of step E is calculated.

$$\mathbf{\Theta}^{(k+1)} = \underset{\mathbf{\Theta}:\sum_c p_c = 1}{\operatorname{argmax}} \; Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}) \tag{24}$$

2.5 Bernoulli mixtures

The Bernoulli classifier can be generalised by using a mixture of several Bernoulli distributions in each class, instead of a single Bernoulli distribution per class. A Bernoulli mixture of I independent components is defined by a set of parameters of the form (15) with

$$\mathbf{\Theta_i}' = \mathbf{p_i} = \begin{pmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_{iD} \end{pmatrix} \in [0,1]^D \tag{25}$$

So $f(\mathbf{x} \mid z_i = 1, \mathbf{p_i})$ is:

$$f(\mathbf{x} \mid z_i = 1, \mathbf{p_i}) = \prod_d {p_{id}}^{x_d} \, (1 - p_{id})^{1 - x_d} \tag{26}$$

The resulting expression of step M to calculate the Bernoulli parameters is:

$${\mathbf{p_i}}^{(k+1)} = \frac{\sum_n {z_{ni}}^{(k)} \mathbf{x_n}}{\sum_n {z_{ni}}^{(k)}} \tag{27}$$

A classifier based on class-conditional Bernoulli mixtures has the form:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \; p(c) \sum_i \left(\, p_{ci} \prod_d \left(\, {p_{cid}}^{x_d} \, (1 - p_{cid})^{1 - x_d} \, \right) \right) \tag{28}$$

2.6 Multinomial mixtures

Multinomial mixtures are analogous to Bernoulli mixtures. A multinomial mixture of I independent components is defined by a set of parameters of the form (15) with

$\mathbf{\Theta_i}'$ defined as in the Bernoulli mixture case (25), and also subject to the additional constraint:

$$\sum_d p_{id} = 1 \tag{29}$$

So $f(\mathbf{x} \mid z_i = 1, \mathbf{p_i})$ is:

$$f(\mathbf{x} \mid z_i = 1, \mathbf{p_i}) = \frac{x_+!}{\prod_d x_d!} \prod_d {p_{id}}^{x_d} \tag{30}$$

The M step in this case is:

$$\mathbf{p_i}^{(k+1)} = \frac{\sum_n {z_{ni}}^{(k)} \mathbf{x_n}}{x_+ \sum_n {z_{ni}}^{(k)}} \tag{31}$$

A classifier based on class-conditional multinomial mixtures has the form:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\ p(c) \sum_i \left(p_{ci} \frac{x_+!}{\prod_d x_d!} \prod_d {p_{cid}}^{x_d} \right) \tag{32}$$

2.7 Multilingual mixtures

Some several extensions of the multinomial text classifiers have been proposed for the case in which the text data is available in two languages. The interest in this task of *bilingual text classification* comes from its potential use in *statistical machine translation*. For example, the problem of learning a complex, global statistical transducer from heterogeneous bilingual sentence pairs can be greatly simplified by first classifying sentence pairs into homogeneous classes and then learning simpler, class-specific transducers.

We begin with a multinomial mixture to compute the probability of a text $p(\mathbf{x})$:

$$p(\mathbf{x}) = \sum_i \alpha_i \, p(\mathbf{x} \mid i) \tag{33}$$

where

$$p(\mathbf{x} \mid i) = \frac{x_+!}{\prod_d x_d!} \prod_d {p_{id}}^{x_d} \tag{34}$$

The goal is to model the probability of a bilingual text pair, $(\mathbf{x}, \mathbf{y})$. To this end, three models have been proposed:

1. *Bilingual bag-of-words model:*

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{z}) \tag{35}$$

where $\mathbf{z}$ is a *bilingual bag-of-words* obtained from the concatenation of the sentences originating $(\mathbf{x}, \mathbf{y})$, and $p(\mathbf{z})$ is a monolingual, multinomial mixture model.

2. *Global (Naive Bayes) decomposition model:*

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})\, p(\mathbf{y}) \tag{36}$$

where $p(\mathbf{x})$ and $p(\mathbf{y})$ are given by (33).

3. *Local (Naive Bayes) decomposition model:*

$$p(\mathbf{x}, \mathbf{y}) = \sum_i \gamma_i\, p(\mathbf{x}, \mathbf{y} \mid i) \tag{37}$$

with
$$p(\mathbf{x}, \mathbf{y} \mid i) = p(\mathbf{x} \mid i)\; p(\mathbf{y} \mid i) \tag{38}$$

where $p(\mathbf{x} \mid i)$ is given by (34) and $p(\mathbf{y} \mid i)$ is an independent multinomial model

$$p(\mathbf{y} \mid i) = \frac{y_+!}{\prod_e y_e!} \prod_e {q_{ie}}^{y_e} \tag{39}$$

The classifier based on (35) has the form:

$$c(x, y) = \underset{c}{\operatorname{argmax}}\ \log p_c + \log \sum_i \alpha_{ci} \prod_d {p_{cid}}^{x_d} \tag{40}$$

In the case of the global decomposition model, it is:

$$c(x, y) = \underset{c}{\operatorname{argmax}}\ \log p_c + \log \sum_i \alpha_{ci} \prod_d {p_{cid}}^{x_d} + \log \sum_i \beta_{ci} \prod_e {q_{cie}}^{y_e} \tag{41}$$

while, in the local decomposition model, we have:

$$c(x, y) = \underset{c}{\operatorname{argmax}}\ \log p_c + \log \sum_i \gamma_{ci} \prod_d {p_{cid}}^{x_d} \prod_e {q_{cie}}^{y_e} \tag{42}$$

3 Applications in Pattern Recognition and Human Language Processing

3.1 OCR using Bernoulli mixture-based classifiers

The OCR task consists in the recognition of handwritten characters or digits from images. Basically, the OCR task is a classification task where we have a class for each

character or digit to recognise. Character or digit images are images that represent binary content, usually black outlines over a white background, so it is a task where a binarisation of the input images is more appropriated to achieve good results. A Bernoulli mixture classifier becomes a good choice to tackle this task.

The OCR task considered consists in the recognition of Indian digits [11], extracted from *courtesy amounts* of real bank drafts. Original samples are given as binary images of different sizes (minimal bounding boxes). To obtain properly normalised images, both in size and position, two simple preprocessing steps were applied. First, each digit image was pasted onto a square background whose centre was aligned with the digit centre of mass. This square background was a white image large enough (64×64) to accommodate most samples though, in some cases, larger background images were required. Second, given a size S, each digit image was subsampled into $S \times S$ pixels, from which its corresponding binary vector of dimension $D = S^2$ was built. Figure 2 shows one preprocessed example of each Indian digit ($S = 30$).

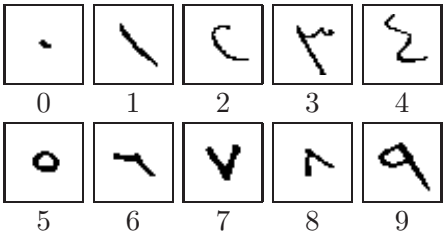


Figure 2: 30×30 examples of each Indian digit.

The standard experimental procedure for classification error rate estimation in the Indian digits task is a simple partition with 7390 samples for training and 3035 for testing (excluding the extra classes *delimiter* and *comma*). Figure 3 shows, for all $S \in \{14, 20, 30\}$ and $I \in \{1, 2, 5, 10, 15, 20, 25\}$, the average error of the I-component Bernoulli mixture classifier tested on the data subsampled at $S \times S$ pixels. Each average was computed from 50 runs of the standard experimental procedure, each run entailing a randomly initialised EM-based learning of a Bernoulli mixture per class. For simplicity, we did not try classifiers with class-conditional mixtures of different number of components; i.e. an I-component classifier means that a mixture of $I_c = I$ Bernoulli components was trained for each digit c.

From the results shown in Figure 3, first note that the curve for $S = 14$ is not as good as those for 20 and 30, which are very similar. Therefore, a subsampling value of 20 can be considered appropriate for this task. Note also that, as expected, the error rate behaviour as a function of I can be described as a smooth concave curve with its minimum at an intermediate value (around $I = 15$). That is, the

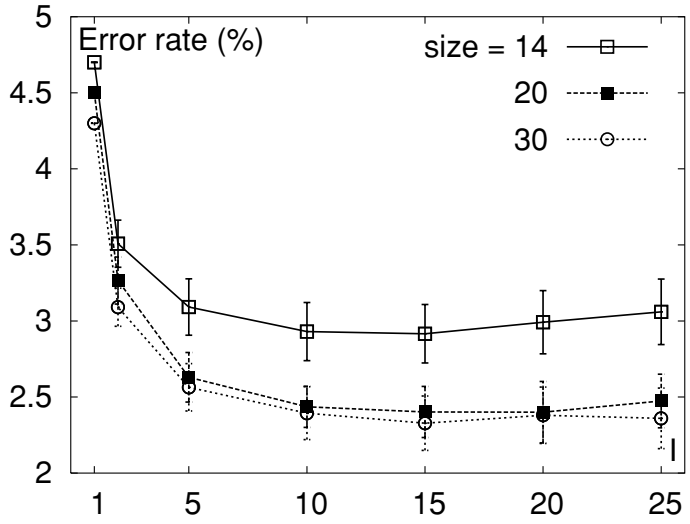


Figure 3: Classification error rate as a function of the number of mixture components in each class (I), for several image sizes. Error bars show standard error.

optimal model complexity is somewhere in between the simplest ($I = 1$) and the most complex ($I \gg$) models.

3.2 Confidence measures in speech recognition

Current speech recognition systems are not error-free and, in consequence, it is desirable for many applications to predict the reliability of each hypothesised word. This can be seen as a conventional pattern recognition problem in which each hypothesised word is to be transformed into a feature vector and then classified as either correct or incorrect. The basic problem then is to decide which predictor (pattern) features and classification model should be used.

As predictor features can be used well known predictors such as: *Acoustic stability*, *Language model probability (LMProb)*, *Hypothesis density (HD)*, *PercPh*, *Duration* and *ACscore*. In addition to these features we have recently introduced "Word Trellis Stability" (WTS) [12]. Other features that we have recently proposed are *WgAC*, *WgLM* and *WgTOT*. These three features are based on word posterior probabilities estimated on multiple word graphs [13].

The classification model used in this task is a naive Bayes model in which parameters are estimated using sophisticated smoothing techniques imported from statistical language modelling [14]. We use $c = 0$ and $c = 1$ for the correct and

incorrect classes, respectively. Given an hypothesised word w and a D-dimensional vector of (discrete) features x, the classification model is:

$$c(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\ p(c \mid w) \prod_{d} p(x_d \mid c, w) \tag{43}$$

We carried out experiments using the *FUB* task, an Italian speech corpus of phone calls to the front desk of a hotel. A training set was used to train Italian context dependent phone models. The acoustic models were left-to-right continuous density HMMS, trained using Linear Discriminant Analysis (LDA) and a Viterbi approximation. Decision-tree clustered generalised triphones were used as phone-units. A smoothed trigram language model was estimated using the transcriptions of the training utterances. The criterion used to measure the performance of the classifier is the *Confidence Error Rate* (CER), defined as the number of classification errors divided by the total number of recognised words. A baseline CER was obtained by assuming that all recognised words are classified as correct. The best results for individual features were given by *WgLM* and *AS*, with a CER of 16.4 and 16.3 respectively. The naive Bayes model was employed to explore the performance of the classifier on many feature combinations. Results are given in Table 3.2.

Features	CER(%)	red.(%)
AcScore + WgTOT_{max} + WTS_{max} + Dur + AS + WgLM_{avg}	13.1	37.6
WTS_{max} + Dur + AS + WgLM_{avg}	13.6	35.2
Dur + AS + WgLM_{max}	14.4	31.4
AS + WgLM_{max}	14.5	31.0
WgLM_{avg}	16.4	21.9
Baseline	21.0	-

Table 1: CER and relative reduction in baseline CER baseline for the best feature combinations

The results show that the single performance is improved through the (naive Bayes) combination of the different features.

3.3 Word disambiguation

Word disambiguation is an interesting problem in rule-based and hybrid machine translation [15]. This problem consists in finding the correct translation of a word in a sentence of a certain *source* language, among all its potential translations into words from a different *target* language. For example, the Spanish word *en* has four

possible translations into Catalan: *en*, *a*, *per* and *amb*. Depending on the particular sentence context where we find *en*, we have to choose its correct translation into Catalan.

Word translation disambiguation of each (ambiguous) word in the input vocabulary entails a separate classification problem, where each possible translation is a class. Using the prefix and suffix (u,v) of the sentence where the ambiguous word appears, we decide the correct translation in accordance with the Bayes decision rule:

$$c(u,v) = \underset{c}{\operatorname{argmax}}\ p(c \mid u,v) \tag{44}$$

To simplify the problem, we assume that word ordering is uninformative, and hence we may represent the context as a "bag of words" $\mathbf{x}$. Also, we limit the context of a word to its immediately surrounding neighbours.

We used the multinomial text classifier for word disambiguation based on the above "bag of words" representation. To avoid overfitting, two smoothing techniques were used: *Laplace smoothing* and *absolute discounting*, where the gained probability mass is distributed among words with null counts (*backing-off*), or all words (*interpolation*), in accordance with a *generalised distribution* such as a *uniform* or *unigram* distribution.

Experiments were based on a parallel Spanish-Catalan corpus extracted from the newspaper *El Periódico*. We used a dictionary with 7085 ambiguous words. Results are shown in Figure 4. They are significantly better than those obtained without the multinomial classifier.

3.4 Text classification

Text classification is a typical task in which multinomial classifiers has been used. Each text (sample) is interpreted as a "bag of words", that is, we assume the order in which words occur in the text is not important. Also, we assume that the number of occurrences of a word does not depend of the number of occurrences of other words (naive Bayes assumption). To improve the comparatively poor performance of the basic, multinomial text classifier, we have proposed two generalisations: the addition of a length model 2.3, and the use of class-conditional multilingual mixtures 2.7.

The addition of the length model was tested on the *BAF* corpus. It is a real task composed of French-English sentences pairs, classified into four classes according to their origin. We used the two smoothing techniques discussed in the previous section. Results are given in Figure 5. From this Figure, it becomes clear that the use of a length model slightly improves the results.

We have done experiments with *BAF* corpus and the bilingual classifiers. The

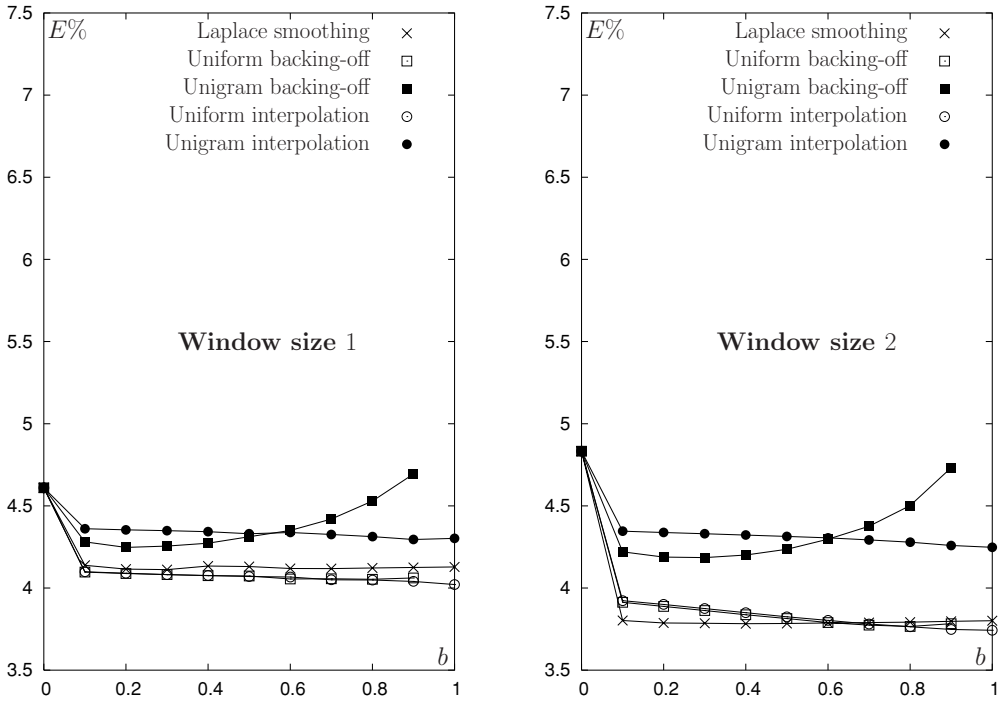


Figure 4: Average error rate (percentage of misclassified contexts) of the multinomial classifier, as a function of the smoothing discount, for several smoothing techniques, window sizes (one in each panel), for both unnormalised and normalised word counts.

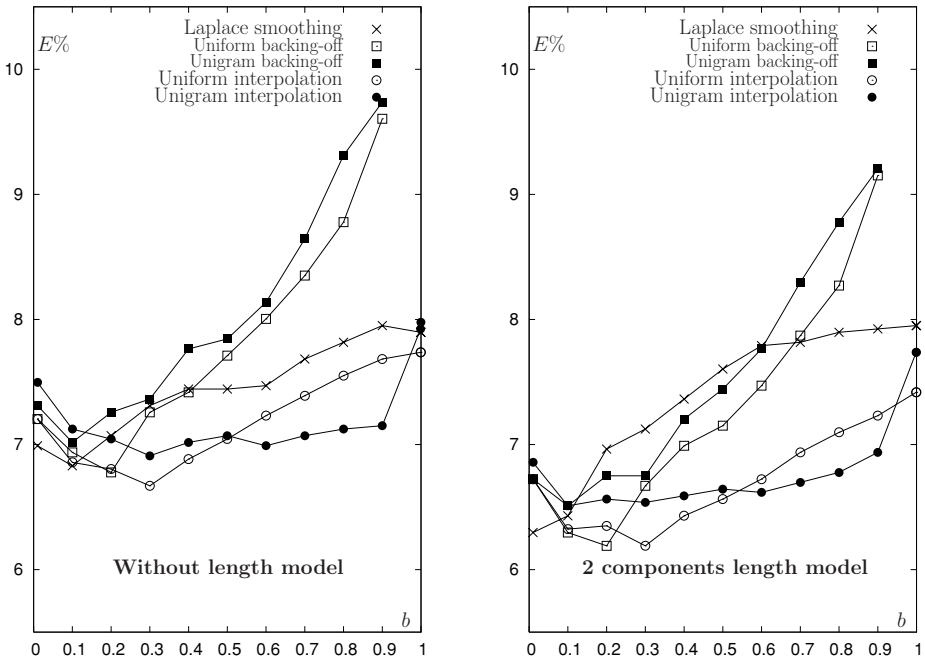


Figure 5: Percentage of bad classified samples in front of the smoothing parameter b for several smoothing and length modelling methods, using the BAF corpus in French.

results can be seen in Figure 6. Two outstanding conclusions can be state from the results shown. First, mixture-based classifiers surpass single-component classifiers in all cases. Second, bilingual classifiers outperform their monolingual counterparts.

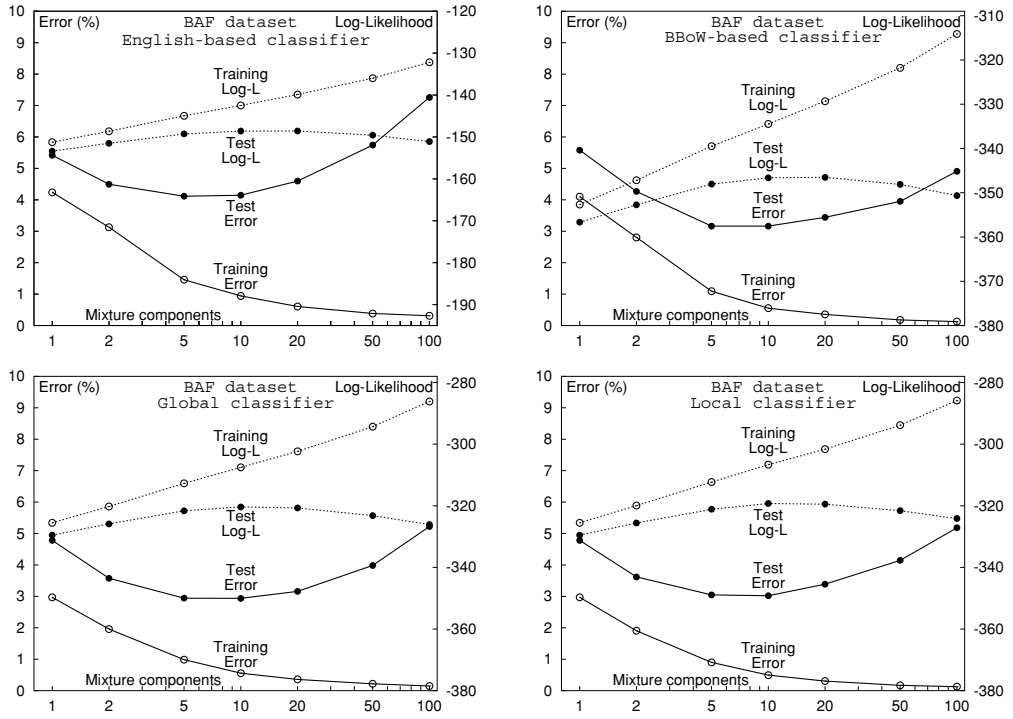


Figure 6: Error rate and log-likelihood curves in training and test sets as a function of the number of mixture components, in *BAF* for the four classifiers considered. Classifiers: the best monolingual, the bilingual bag-of-words (BBoW), the global and the local classifier.

4 Concluding remarks

In this paper, we have reviewed the naive Bayes classifier, its conventional instantiations, and several generalisations and applications studied by the "Pattern Recognition and Human Language Technology" research group. In particular, we have reviewed the Bernoulli and multinomial instantiations, a version of the multinomial classifier enriched by a length model, and extensions of the two basic instantiations to class-conditional mixtures. Also, we reviewed extensions of the class-conditional

multinomial mixture-based classifier to the case of bilingual data. Regarding applications, we have described successful application of these models to very different tasks: OCR, confidence measures for speech recognition, word disambiguation in machine translation and general text classification.

References

[1] A. Juan and E. Vidal. On the use of Bernoulli mixture models for text classification. *Pattern Recognition*, 35(12):2705–2710, 2002.

[2] K. Nigam et al. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[3] J. Novovicová and A. Malík. Application of Multinomial Mixture Model to Text Classification. In *Proc. of IbPRIA 2003*, pages 646–653, 2003.

[4] David Vilar et al. Effect of Feature Smoothing Methods in Text Classification Tasks. In *Proc. of PRIS'04*, pages 108–117, 2004.

[5] D. Pavlov et al. Document Preprocessing For Naive Bayes Classification and Clustering with Mixture of Multinomials. In *Proc. of KDD'04*, pages 829–834, 2004.

[6] F. Peng et al. Augmenting Naive Bayes classifiers with statistical language models. *Information Retrieval*, 7(3):317–345, 2003.

[7] J. Rennie et al. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proc. of ICML'03*, pages 616–623, 2003.

[8] Tobias Scheffer and Stefan Wrobel. Text Classification Beyond the Bag-of-Words Representation. In *Proc. of ICML'02 Workshop on Text Learning*, 2002.

[9] A. Giménez, J. Andrés, and A. Juan. Modelizado de la longitud para clasificación de textos. In Nicolás Pérez de la Blanca Capilla and Filiberto Plà Bañón, editors, *Actas del I Workshop sobre Reconocimiento de Formas y Análisis de Imágenes (CEDI 2005), Simposio de la Asociación Española de Reconocimiento de Formas y Análisis de Imágenes (AERFAI). ISBN: 84-9732-445-5*, pages 21–28, Granada (Spain), September 12-13 2005. Thomson.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[11] Alfons Juan and Enrique Vidal. Bernoulli mixture models for binary images. In *Proc. of the 17th Int. Conf. on Pattern Recognition (ICPR 2004)*, volume 3, Cambridge (UK), August 2004.

[12] A. Sanchis, A. Juan, and E. Vidal. Estimating confidence measures for speech recognition verification using a smoothed naive bayes model. In Francisco José Perales, Aurélio J. C. Campilho, Nicolás Pérez de la Blanca, and Alberto Sanfeliu, editors, *Pattern Recognition and Image Analysis, First Iberian Conference IbPRIA 2003 Proceedings*, Lecture Notes in Computer Science LNCS 2652, pages 910–918, Port d'Andratx, Mallorca, Spain, jun 2003. Springer-Verlag.

[13] A. Sanchis, A. Juan, and E. Vidal. New features based on multiple word graphs for utterance verification. In *8th International Conference on Spoken Language Processing*, pages 2545–2548, October 2004.

[14] A. Sanchis, A. Juan, and E. Vidal. Improving utterance verification using a smoothed naive bayes model. In *IEEE International Conference on Acoustic, Speech and Signal Processing*, volume 1, pages 592–595. IEEE Press, April 2003.

[15] J. Andrés, J. Navarro, A. Juan, and F. Casacuberta. Word translation disambiguation using multinomial classifiers. In *Iberian Conference on Pattern Recognition and Image Analysis*, volume 3523 of *Lecture Notes in Computer Science*, pages 622–629. Springer-Verlag, Estoril (Portugal), June 2005.

[16] Alfons Juan, José García-Hernández, and Enrique Vidal. EM Initialisation for Bernoulli Mixture Learning. In A. Fred et al., editor, *Proc. of the SSPR-SPR04, LNCS 3138*, Lecture Notes in Computer Science, 3138, pages 635–643, Lisbon (Portugal), August 2004. Springer.

[17] José García-Hernández, Vicent Alabau, Alfons Juan, and Enrique Vidal. Bernoulli mixture-based classification. In A. R. Figueiras-Vidal et al., editor, *Proc. of the LEARNING04, ISBN 84-688-8453-7*, Elche (Spain), October 2004.

Audiovisual biometric verification

José Luis Alba-Castro, Carmen García-Mateo,
Daniel González-Jiménez, Enrique Argones-Rúa
University of Vigo. Signal Processing Group.
ETSETelecomunicación, Campus Universitario de Vigo

Abstract

Single-mode biometric verification systems differ each other greatly in terms of performance and vulnerability to spoofing. Face and Voice identification technology are not as accurate as others, such as iris, retina or fingerprint scanning, but have features that beat other biometric options in a multitude of practical scenarios and they can be the preferable option in wide-spread applications like remote accessing to Internet. Audiovisual biometric techniques try to improve accurateness by merging face and voice traits in such a way that False Rejection and False Acceptance Rates are always smaller than using the alternative single-mode biometric systems alone. In this work we present the advances of our group in face and voice verification. We also present an internet-based application for secure identity authentication using audiovisual biometric patterns.

Keywords: biometric features, face and voice-based recognition, web-based authentication.

1 Introduction

Classical techniques for electronic person authentication have several drawbacks in terms of performing reliable and user-friendly identity recognition; this occurs particularly with remote operations, more prone to hacker attacks. Automatic identity verification, based on distinctive anatomical features (e.g., face, voice, fingerprint, iris, etc.) and behavioral characteristics (e.g., online/offline signature, keystroke dynamics, etc), is becoming an increasingly reliable standalone solution and attracting a great deal of attention as far as remotely-based applications are concerned. Some of the biometric-inherited drawbacks associated with large-scale deployment of any biometric authentication application can be partially circumvented using simultaneous or alternative biometric traits [1] that mitigate the problems associated with spoofing, failure-to-enroll, noise in a particular sensor or acquired feature, intra-class variation and inter-class similarities.

Nowadays, face and voice are the only biometric traits that can be captured at very low cost in almost any desktop, laptop or cellular-phone in the market. It is true that other biometric traits, like fingerprints, iris or even behavioral traits like signature dynamics, can be captured with desktop devices under 200 euros and are acknowledged as being more robust and accurate than audiovisual features (at least fingerprints and iris), but people is much more concerned about these patterns being captured and handled by automatic systems.

Edited by F.Pla, P.Radeva, J.Vitrià, 2006.

When talking about widespread internet-based applications these drawbacks make audiovisual biometric traits the preferable choice for a multi-biometric authentication system.

In this paper we present the advances of our research group on face and voice authentication and give results on two well-known multi-modal databases: XM2VTS [2] and BANCA [3]. The algorithms shown in this work are integrated in an open framework for distributed biometric authentication in a web environment [4], that will be superficially described also in this paper.

The paper is organized as follows: section 2 is dedicated to advances in face authentication based on local matching approaches. Several methods to combine local information are explained and results given for the two mentioned databases. Section 3 details the state of the art speaker recognition algorithms used for authentication and shows results on BANCA database. Section 4 is dedicated to summarize the main features of the web-based biometric authentication system and section 5 gives some conclusions and future research lines.

2 Face verification through local matching approaches

One of the most successful approaches to automatic face recognition is the Elastic Bunch Graph Matching algorithm (EBGM) [5]. It combines local and global representation of the face by computing multi-scale and multi-orientation Gabor responses (jets) from a set of the so-called fiducial points, located at specific face regions (eyes, tip of the nose, mouth..., i.e. "universal" features). Finding every fiducial point relies on a matching process between the candidate jet and a bunch of jets extracted from the corresponding fiducial points of different faces. This matching problem is solved by maximizing a function that takes texture and geometrical distortion into account. In this way, there are several variables that can affect the accuracy of the final positions, as differences in pose, illumination conditions and insufficient representativeness of the stored bunch of jets. Once fiducial points are adjusted, only textural information (Gabor jets) is used in the classifier.

The main differences between EBGM and our approach [6] are focused on the way we locate and match fiducial points and on the final dissimilarity function that does not only use texture but also geometrical information. Our method locates salientable points in face images by means of the ridges and valleys operator. As only some basic image operations are needed, the computational load is reduced from the original EBGM algorithm and, at the same time, possible discriminative locations are found in an early stage of the recognition process. In this sense we say that this method is inherently discriminative, in contrast to trainable parametric models. The set of selected points turned out to be quite robust against illumination conditions and slight variations in pose. Many of the located fiducial points belong to "universal" features, but many others are person-dependent. So, EBGM locates a pre-defined set of "universal" features and our approach finds a person-dependent set of features. The correspondence between fiducial points of two faces only uses geometrical information and it is based on shape contexts [7]. As a byproduct of the correspondence algorithm, we extract measures of local geometrical distortion. Gabor jets are also calculated from the adjusted points and the final dissimilarity function compiles geometrical and textural information.

2.1 Shape-driven point selection

In this work, shape information has been obtained using the ridges and valleys operator because of its robustness against illumination changes [8]. Moreover, the relevance of valleys in face shape description has been pointed out by some cognitive science works [9]. In this paper, we have used the ridges and valleys obtained by thresholding the so-called multi local level set extrinsic curvature (MLSEC) [10]. The MLSEC operator works here as follows: **i)** computing the normalized gradient vector field of the smoothed image, **ii)** calculating the divergence of this vector field, which is bounded and gives an intuitive measure of valleyness (positive values running from 0 to 2) and ridgeness (negative values from -2 to 0), and **iii)** thresholding the response so that image pixels where the MLSEC response is smaller than -1 are considered ridges, and those pixels larger than 1 are considered valleys.

Once the feature descriptor has been properly defined, we have a way of describing fiducial points in terms of positions where the geometrical image features have been detected. For this shape descriptor to be useful in face recognition or authentication, local texture information must be also taken into account. Gabor wavelets are biologically motivated convolution kernels that capture this kind of information and are also quite invariant to the local mean brightness, so an efficient face encoding approach will be to extract texture from these geometrically salience regions.

Figure 1: Left: Original Image. Center-left: Valleys and ridges image. Center-right: Thresholded ridges image. Right: Thresholded valleys image

After ridges and valleys in a new image have been extracted, we must sample these lines in order to keep a set of points for further processing. There are some possible combinations, in terms of using just ridges, just valleys or both of them, so we will refer to the binary image, obtained as a result of the previous step, as the sketch from now on.

In order to select a set of points from the original sketch, a dense rectangular grid ($\mathcal{N}_x \times \mathcal{N}_y$ nodes) is applied onto the face image and each grid node changes its position until it finds the nearest line of the sketch. So, finally, we get a vector of points $\mathcal{P} = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_n}\}$[1], where $\vec{p_i} \in \mathbb{R}^2$. These points sample the original sketch, as it can be seen in figure 2.

[1] $n = \mathcal{N}_x \times \mathcal{N}_y$. Typical sizes for n are 100 or more nodes

2.2 Point Matching

Once we have the two face images, $\mathcal{F}_1$ and $\mathcal{F}_2$, at common size, we want to proceed to compute similarity between them. Let $\mathcal{P} = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_n}\}$ be the set of points for $\mathcal{F}_1$, and $\mathcal{Q} = \{\vec{q_1}, \vec{q_2}, \ldots, \vec{q_n}\}$ the set of points for $\mathcal{F}_2$.

In order to compare feature vectors extracted at these positions, we must first compute the matching between points from both images. We have adopted the idea described in [7]. For each point i in the constellation, we compute a 2-D histogram h_i of the relative position of the remaining points, so that a vector of distances $\mathcal{D} = \{d_{i1}, d_{i2}, \ldots, d_{in}\}$ and a vector of angles $\vec{\theta} = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}\}$ are calculated for each point. As in [7], we employ bins that are uniform in log-polar space, i.e. the logarithm of distances is computed. Each pair $(\log d_{ij}, \theta_{ij})$ will increase the number of counts in the adequate bin of the histogram.

Once the sets of histograms are computed for both faces, we must match each point in the first set $\mathcal{P}$ with a point from the second set $\mathcal{Q}$. A point $\vec{p}$ from $\mathcal{P}$ is matched to a point $\vec{q}$ from $\mathcal{Q}$ if the term C_{pq}, defined as:

$$C_{pq} = \sum_k \frac{\left[h_p(k) - h_q(k)\right]^2}{h_p(k) + h_q(k)} \tag{1}$$

is minimized[2]. Finally, we have a correspondence between points defined by ξ:

$$\xi(i) : \vec{p_i} \Longrightarrow q_{\vec{\xi(i)}} \tag{2}$$

where $\vec{p_i} \in \mathcal{P}$ and $q_{\vec{\xi(i)}} \in \mathcal{Q}$.

2.3 Local texture matching through Gabor jets similarities

The system uses a set of 40 Gabor filters, with the same configuration employed in [5]. These filters are convolution kernels in the shape of plane waves restricted by a Gaussian envelope,

[2] k in (1) runs over the number of bins in the 2D histogram

Figure 2: Left: Original rectangular dense grid. Center: Valleys and ridges sketch. Right: Grid adjusted to the sketch.

as it is shown next:

$$\psi_m\left(\overrightarrow{x}\right) = \frac{\left\|\overrightarrow{k}_m\right\|^2}{\sigma^2} \exp\left(-\frac{\left\|\overrightarrow{k}_m\right\|^2 \left\|\overrightarrow{x}\right\|^2}{2\sigma^2}\right)\left[\exp\left(i\,\overrightarrow{k}_m \cdot \overrightarrow{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{3}$$

where $\overrightarrow{k}_m$ contains information about frequency and orientation of the filters, $\overrightarrow{x} = (x, y)^T$ and $\sigma = 2\pi$.

The region surrounding a pixel in the image is encoded by the convolution of the image patch with these filters, and the set of responses is called a jet, $\mathcal{J}$. So, a jet is a vector with 40 coefficients, and it provides information about a specific region of the image. At point $\vec{p}_i = [x_i, y_i]^T$, we get the following feature vector:

$$\{\mathcal{J}_{\vec{p}_i}\}_m = \sum_x \sum_y I(x, y)\psi_m\left(x_i - x, y_i - y\right) \tag{4}$$

where $\{\mathcal{J}_{\vec{p}_i}\}_m$ stands for the m-th coefficient of the feature vector extracted from $\vec{p}_i$. The textural score between two images is:

$$\mathcal{S}_{\mathcal{J}} = f_n\left\{< \mathcal{J}_{\vec{p}_i}, \mathcal{J}_{\vec{q}_{\xi(i)}} >\right\}_{\vec{p}_i \in \mathcal{P}} \tag{5}$$

where $< \mathcal{J}_{\vec{p}_i}, \mathcal{J}_{\vec{q}_{\xi(i)}} >$ represents the normalized dot product between correspondent jets, but taking into account that only the moduli of jet coefficients are used. In (5), f_n stands for a generic combination rule of the n dot products.

2.4 Shape distortion as dissimilarity measurement

Once we have extracted the ridges and valleys from two face images, a global shape score can be obtained. One of the most successful dissimilarity measurements for sets of points (or binary images) is the Hausdorff distance, that has been widely used for object matching in scene analysis [11]. It is well known that the standard Hausdorff distance is quite sensible to outliers, so some modifications [12] have been used to avoid such a problem. In this work, we have used a particular modification that can be referred as *Average Hausdorff Distance (AHD)*. Given two sets $\mathcal{A}$ and $\mathcal{B}$, the directed Average Hausdorff Distance $ahd(\mathcal{A}, \mathcal{B})$ from the set $\mathcal{A}$ to the set $\mathcal{B}$, (assuming Euclidean distance between set elements) is:

$$ahd(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \min_{b \in \mathcal{B}}(\|a - b\|) \tag{6}$$

where $|\mathcal{A}|$ denotes the cardinal of the set $\mathcal{A}$. So, the (symmetric) *Average Hausdorff Distance* (AHD) can be formally written as:

$$AHD(\mathcal{A}, \mathcal{B}) = \frac{1}{2}(ahd(\mathcal{A}, \mathcal{B}) + ahd(\mathcal{B}, \mathcal{A})) \tag{7}$$

The computation of $AHD(\mathcal{A}, \mathcal{B})$ is easily performed as a double dot product: given our binary image $\mathcal{F}_1(x, y)$ that can be thought of as the output of any contour operator, with

$\mathcal{A} = \{(x,y)|\mathcal{F}_1(x,y) = 1\}$; we can define $\overrightarrow{\mathcal{F}_1}$ as the binary vector associated to the binary image $\mathcal{F}_1$, and $\widehat{\mathcal{F}_1} = \frac{1}{|\mathcal{A}|}\overrightarrow{\mathcal{F}_1}$ the associated normalized vector. For a digital binary image, we can define the Distance Transform, $D(\mathcal{F}_1)$ [13], as a point-wise transform that contains, for each pixel, the distance between that pixel and the pixel of value 1 closest to it. The vector format for the distance transform $D(\overrightarrow{\mathcal{F}_1})$ can also be extended to the associated normalized image $D(\widehat{\mathcal{F}_1})$ with the same meaning. With these definitions, the AHD between binary images can then be calculated averaging inner products:

$$AHD(\mathcal{F}_1,\mathcal{F}_2) = \frac{1}{2}(< \widehat{\mathcal{F}_1}, D(\widehat{\mathcal{F}_2}) > + < \widehat{\mathcal{F}_2}, D(\widehat{\mathcal{F}_1}) >) \tag{8}$$

Now that global shape distortion has been taken into account throughout the computation of $AHD(\mathcal{F}_1,\mathcal{F}_2)$, local shape distortions will be handled. So, we introduce two different terms here:

$$\mathcal{GD}_1(\mathcal{F}_1,\mathcal{F}_2) \equiv \mathcal{GD}_1(\mathcal{P},\mathcal{Q}) = \sum_{i=1}^{n} v_i C_{i\xi(i)} \tag{9}$$

$$\mathcal{GD}_2(\mathcal{F}_1,\mathcal{F}_2) \equiv \mathcal{GD}_2(\mathcal{P},\mathcal{Q}) = \sum_{i=1}^{n} w_i \left\|\overrightarrow{p_i c_{\mathcal{P}}} - \overrightarrow{q_{\xi(i)} c_{\mathcal{Q}}}\right\| \tag{10}$$

Equation (9) computes geometrical distortion by linearly combining the individual costs represented in (1). On the other hand, (10) calculates metric deformation by combining the norm of the difference vector between matched points[3].

Weighting vectors v and w can be simply set to the vector $\overrightarrow{1}$ or can be discriminatively calculated. When dealing with face shape distortion, it is obvious that regions related to face muscles are more likely to suffer slight displacements than others. Hence, the local contributions in $\mathcal{GD}_1$ and $\mathcal{GD}_2$ must be weighted accordingly. We have found the n components of v and w as the Fisher best discriminative direction between the local shape distortion vectors for evaluation clients and impostors. $\mathcal{GD}_1$ and $\mathcal{GD}_2$ can be seen as global shape distortion measurements, that should be large for faces of different subjects and small for faces representing the same person. If faces are in an upright position and are scaled at the same size, adding the global distortion $AHD(\mathcal{F}_1,\mathcal{F}_2)$ increases the discriminative power of the shape part of the classifier, as it will be seen at the results section.

Now we can think of linearly combining jet dissimilarity, $[1 - \mathcal{S}_{\mathcal{J}}(\mathcal{F}_1,\mathcal{F}_2)]$, with shape deformations, resulting in the final dissimilarity function $\mathcal{DS}(\mathcal{F}_1,\mathcal{F}_2)$:

$$\mathcal{DS}(\mathcal{F}_1,\mathcal{F}_2) = \lambda_1[1 - \mathcal{S}_{\mathcal{J}}(\mathcal{F}_1,\mathcal{F}_2)] + \lambda_2\mathcal{GD}_1(\mathcal{F}_1,\mathcal{F}_2) + \lambda_3\mathcal{GD}_2(\mathcal{F}_1,\mathcal{F}_2) + \lambda_4 AHD(\mathcal{F}_1,\mathcal{F}_2) \tag{11}$$

with $\lambda_i > 0$. The combination of $\mathcal{GD}_1$ and $\mathcal{GD}_2$ is what we call *Sketch Distortion* (SKD). If we set $f_n \equiv mean$, from (11) and using (5), (9) and (10), it follows that $\mathcal{DS}(\mathcal{F}_1,\mathcal{F}_2)$ is equal to:

$$\sum_{i=1}^{n}\left[\lambda_1\frac{1 - < \mathcal{J}_{\vec{p}_i}, \mathcal{J}_{\vec{q}_{\xi(i)}} >}{n} + \lambda_2 C_{i\xi(i)} + \lambda_3\left\|\overrightarrow{p_i c_{\mathcal{P}}} - \overrightarrow{q_{\xi(i)} c_{\mathcal{Q}}}\right\|\right] + \lambda_4 \cdot AHD(\mathcal{F}_1,\mathcal{F}_2) \tag{12}$$

[3]Note that the centroid of the constellation has been subtracted from the point coordinates in order to deal with translation

Table 1: Sketch Distortion (SKD) between the face images from figures 3 to 4

		Subject A		Subject B	
		Image 1	Image 2	Image 1	Image 2
Subject A	Image 1	0	1851	3335	3226
	Image 2	1851	0	3053	2821
Subject B	Image 1	3335	3053	0	1889
	Image 2	3326	2821	1889	0

In (12) we can see that each contribution of jet dissimilarity is modified with a weighted geometrical distortion (the so-called *Local Sketch Distortion* or $LSKD$). A high value in $LSKD$ from the pair $\left(\vec{p_i}, q_{\vec{\xi(i)}}\right)$ means that they are not positioned over the same face region, so that jet dissimilarity will also be high. This fact is more likely to occur when incoming faces do not represent the same person. Even if $LSKD$ is low, but faces do not belong to the same person, textural information will increase the dissimilarity between them. On the other hand, when faces belong to the same subject, low $LSKD$ values should be generally achieved, so that matched points are located over the same face region, resulting in a low jet dissimilarity. Thus, the measurement in (12) reinforces discrimination between subjects. Figures 3 and 4 give a visual understanding of this concept. Figure 3 shows two instances of face images from subject A, while faces in figure 4 belong to subject B. The visual geometric difference between the two persons is reflected in the Sketch Distortion term, whose values are shown in table 1.

The scores weighting vector $\overrightarrow{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T$ is absolutely necessary to avoid that scores with weak performance provoke an useless score combination.

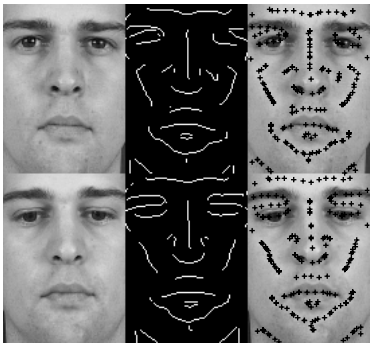


Figure 3: **Top**: Left: First image from subject A. Center: Valleys and ridges sketch. Right: Grid adjusted to the sketch. **Bottom**: Left: Second image from subject A. Center: Valleys and ridges sketch. Right: Grid adjusted to the sketch.

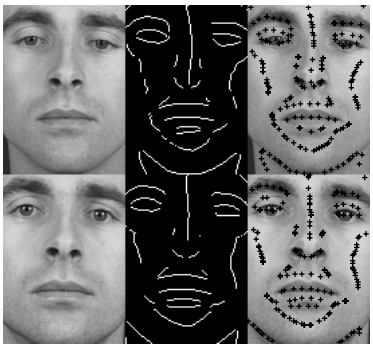


Figure 4: **Top**: Left: First image from subject B. Center: Valleys and ridges sketch. Right: Grid adjusted to the sketch. **Bottom**: Left: Second image from subject B. Center: Valleys and ridges sketch. Right: Grid adjusted to the sketch.

2.4.1 Results over the XM2VTS database

We tested our method using the XM2VTS database on configuration I of the Lausanne protocol [14]. The XM2VTS database contains synchronized image and speech data recorded on 295 subjects during four sessions taken at one month intervals. The database was divided into three sets: a training set, an evaluation set, and a test set. The training set was used to build client models, while the evaluation set was used to select the most discriminative nodes and to estimate thresholds. Finally, the test set was only used to measure the performance of the system. The 295 subjects were divided into a set of 200 clients, 25 evaluation impostors, and 70 test impostors. The results are presented in table 2. In the first row of this table, although only textural information (T) is used, i.e. $\lambda_1 = 1, \lambda_{2,3,4} = 0$, some shape information still remains, because jets are extracted and compared at geometrically matched fiducial points. The next row shows the performance using only the AHD score. Rows 3rd and 4th show the performance using the $\mathcal{GD}_1$ and the $\mathcal{GD}_2$ scores with Fisher weighting vectors (v and w) for balancing local shape distortion. The results in the fifth row ($T + SKD$) were achieved by using $\lambda_{1,2,3} = 1, \lambda_4 = 0$. Sixth row ($T + AHD$) shows performance with $\lambda_{1,4} = 1, \lambda_{2,3} = 0$. Next row presents the error rates with $\overrightarrow{\lambda} = [1, 1, 1, 1]^T$. Finally, the last row shows the results using the two vectors v and w mentioned above, and a second level of Fisher discriminative weighting for balancing individual scores λ_i.

¿From this table we can highlight: **i)** Textural information extracted from person-dependent points performs better than any of the shape measurements tested, **ii)** $\mathcal{GD}_1$ and $\mathcal{GD}_2$, obtained as a byproduct of the point matching process do not perform well alone. Moreover, the direct combination of SKD with jet dissimilarity yields a worse performance than using Gabor responses alone, and the same for ($T+AHD$) and ($T+SKD+AHD$), but **iii)** both types of shape distortion help to reduce error rates when they are discriminatively combined with jet dissimilarity (a relative improvement of 13.53%).

Up to now, all feature vectors (jets) have been weighted equally to get the texture score, i.e. $\mathcal{S}_{\mathcal{J}}$. In the following sections we will explain two different strategies to weigh and select

Table 2: $FRR_{ev}(\%)$,$FAR_{ev}(\%)$,$FAR_{test}(\%)$ and $FRR_{test}(\%)$ (at EER threshold) for different configurations

Method	$FRR_{ev}(\%)$	$FAR_{ev}(\%)$	$FAR_{test}(\%)$	$FRR_{test}(\%)$
Textural (T)	3.17	2.36	2.5	5.11
AHD	8.67	6.08	11.75	7.22
$\mathcal{GD}_1$	13.5	6.41	29.75	11.12
$\mathcal{GD}_2$	13.17	7.21	38	12.09
$T + SKD$	3.33	1.73	5.75	4.23
$T + AHD$	4.17	2.76	4.75	4.93
$T + SKD$+AHD	2.67	2.02	4.25	4.26
Fisher combination	1.83	1.86	2.25	4.33

Figure 5: Rectangular grid used to take the local features

feature vectors from a given image. The first of them is a LDA-based approach used when Gabor jets are extracted from nodes in a rectangular grid. The other one is an accuracy-based method, and it can be applied to rectangular grids or to select the best nodes obtained through Ridges and Valleys sampling.

2.5 Combining local similarities

In this section we show a strategy [15] to weight the different Gabor similarities when the Gabor jets are located in small windows which are centered following the rectangular grid pattern that we can see in the figure 5. The face images have been normalized to align the center of the eyes and the mouth to the same windows for all the images. This grid has 13 rows and 10 columns, so we have $N = 130$ Gabor jets with 40 coefficients each encoding every frontal face image.

Let $\mathcal{P} = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_N}\}$ denote the set of points we use to extract the texture information, and $\mathcal{J} = \{\mathcal{J}_{\vec{p_1}}, \mathcal{J}_{\vec{p_2}}, \ldots, \mathcal{J}_{\vec{p_N}}\}$ be the set of jets calculated for one face. The similarity function between two Gabor jets taken from two different images $\mathcal{I}^1$ and $\mathcal{I}^2$ results in:

$$\mathcal{S}\left(\mathcal{J}^1_{\vec{p_i}}, \mathcal{J}^2_{\vec{p_i}}\right) =< \mathcal{J}^1_{\vec{p_i}}, \mathcal{J}^2_{\vec{p_i}} > , \tag{13}$$

where $< \mathcal{J}^1_{\vec{p_i}}, \mathcal{J}^2_{\vec{p_i}} >$ represents the normalized dot product between the i-th component from $\mathcal{J}^1$ and the corresponding component from $\mathcal{J}^2$, but taking into account that only the moduli of jet coefficients are used.

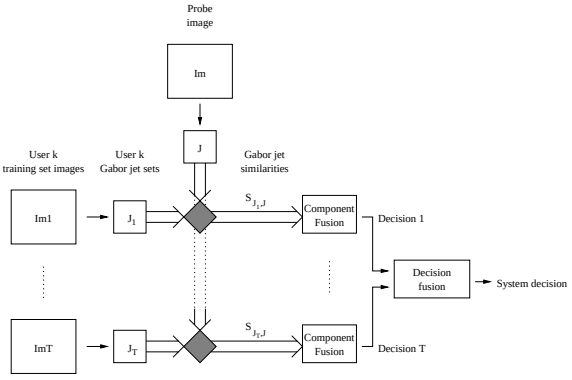


Figure 6: Decision-fusion scheme

So, if we want to compare two frontal face images, we will get, using the equation 13, the following similarity set:

$$\mathcal{S}_{\mathcal{I}^1,\mathcal{I}^2} = \{\mathcal{S}\left(\mathcal{J}^1_{\vec{p1}}, \mathcal{J}^2_{\vec{p1}}\right), \ldots, \mathcal{S}\left(\mathcal{J}^1_{\vec{pN}}, \mathcal{J}^2_{\vec{pN}}\right)\} \tag{14}$$

These similarity scores then have to be combined to a single decision score output by an appropriate fusion rule.

When we have T training images for the client training we have several choices. One of them is to make a decision based on the similarity set that we can get comparing a single *user template* with the probe image. On the other hand we could use the Gabor jets of every training image as a template, and then obtain T different decision scores. This approach, which is the information fusion approach adopted in this paper and is referred as *multiple template* method, then requires the fusion of decision scores corresponding to the individual templates.

2.5.1 Information Fusion

Let us suppose that we have T different training images for every client. We can then build a set of T decision functions for the user k, and we can write them as:

$$\mathcal{D}^k_i\left(\mathcal{J}\right) = f\left(\mathcal{J}, \mathcal{J}^{k,i}\right), i \in \{1, \ldots, T\}\,, \tag{15}$$

where $\mathcal{J}^{k,i}$ denotes the i^{th} training image for user k, and assuming that the decision functions $f\left(\cdot\right)$ computed for the respective training images are identical.

As indicated in the previous Section, the decision function $\mathcal{D}^k_i\left(\mathcal{J}\right)$ is realised as a two step operation where by in the first step we obtain similarity scores for the individual local jets and in the second stage we fuse these scores by a fusion rule, $g(\cdot)$, i.e.

$$f\left(\mathcal{J}, \mathcal{J}^{k,i}\right) = g\{\mathcal{S}\left(\mathcal{J}_{\vec{p1}}, \mathcal{J}^{k,i}_{\vec{p1}}\right), \ldots, \mathcal{S}\left(\mathcal{J}_{\vec{pN}}, \mathcal{J}^{k,i}_{\vec{pN}}\right)\} \tag{16}$$

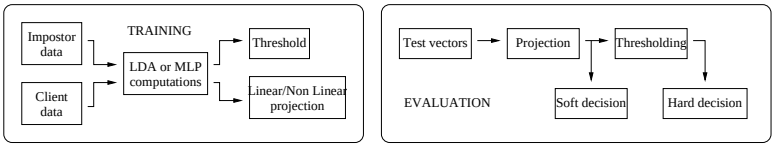


Figure 7: LDA or MLP based fusion

The decision scores obtained for the multiple templates then have to be fused. The decision fusion function can be defined as $\mathcal{D}^k(\mathcal{D}_1^k, \ldots, \mathcal{D}_T^k)$, and can be performed by any suitable fusion function such as those described in the next Section 2.5.2. This decision fusion function must take the final decision about the identity claim as

$$\mathcal{D}^k = h\left(D_1^k, \ldots, D_T^k\right) \tag{17}$$

An overview of the scheme is shown in figure 6.

2.5.2 Fusion Methods

The fusion of image component similarity scores defined in equation 16 as well as the decision score fusion in equation 17 can be implemented using one of several trainable or non trainable functions or rules for this task, as MLP, SVM, LDA, AdaBoost or the sum rule. For this experiment we will compare the performance of MLP and LDA. In figure 7 we can see an overview of the training and evaluation processes with these methods. Both LDA and MLP outputs are not thresholded in the decision score level because it could cause a loss of information in this stage.

The MLP that we use in this experiment is a fully connected and one hidden layer network. Based on some previous work we decided to use 3 neurons in the hidden layer to get the decision scores and 2 neurons in the hidden layer for the decision score fusion. We have trained the MLPs using the standard backpropagation algorithm.

2.5.3 LDA-based Feature Selection

In a two class problem, LDA yields just one direction vector. Each component v_i of the LDA vector $\mathbf{v}$ represents the weight of the contribution of the i^{th} component to the separability of the two classes as measured by the eigenvalue of the LDA eigenanalysis problem. At this point it is pertinent to ask whether the coefficient values could be used to judge which of the features are least useful from the point of view of class separation. If there was a basis for identifying irrelevant features, we could reduce the dimensionality of the problem and at the same time improve the performance of the fusion system. This is the normal positive outcome one can expect from feature selection.

To answer this question, let us look at the LDA solution in more detail. Let $\mathcal{X} = [x_1, \ldots, x_N]$ denote our Gabor jet similarities vector. Clearly, x_i are not independent, as ideally, all similarity values should be high for the true identity claim and vice-versa for

an impostor claim. However, it is not unreasonable to assume that x_i is class conditional independent of x_j $\forall i, j | i \neq j$ and $i, j \in \{1, \ldots, N\}$. This is a relatively strong assumption, but for the sake of simplicity, we shall adopt it.

Let the mean of the i^{th} component be denoted $\mu_{i,0} = E\{x_i | C = 0\}$ and $\mu_{i,1} = E\{x_i | C = 1\}$, where $C = 1$ when $\mathcal{X}$ comes from a true identity claim and $C = 0$ when $\mathcal{X}$ comes from a false identity claim. Let $\mu_i = \frac{1}{2}(\mu_{i,0} + \mu_{i,1})$. Further, let $\sigma^2_{i,0} = \{(x_i - \mu_{i,0})^2 | C = 0\}$ and $\sigma^2_{i,1} = \{(x_i - \mu_{i,1})^2 | C = 1\}$ denote the variances of the similarity scores. Let $c_i = \frac{1}{2}(\sigma^2_{i,0} + \sigma^2_{i,1})$.

As x_i represents similarity and the greater the similarity the higher the value of x_i, we can assume $\mu_{i,1} > \mu_{i,0}, \forall i \in \{1, \ldots, N\}$.

LDA finds a one dimensional subspace in which the separability of true clients and impostors is maximised. The solution is defined in terms of the within class and between class scatter matrices S_w and S_b respectively, i.e.

$$S_w = \begin{pmatrix} c_1 & 0 & \ldots & 0 \\ 0 & c_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & c_N \end{pmatrix} \tag{18}$$

$$S_b = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^T \tag{19}$$

where μ_C is the mean vector of class C composed of the above components.

Now the LDA subspace is defined by the solution to the eigenvalue problem

$$S_w^{-1} S_b \mathbf{v} - \lambda \mathbf{v} = 0 \tag{20}$$

In our face verification case equation 20 has only one non zero eigenvalue λ and the corresponding eigenvector defines the LDA subspace. It is easy to show that the eigenvector $\mathbf{v}$ is defined as

$$\mathbf{v} = S_w^{-1}(\mu_1 - \mu_0) \tag{21}$$

Recall that all the components of the difference of the two mean vectors are non negative. Then from equations 21 and 18 it follows that the components of the LDA vector $\mathbf{v}$ should also be non negative. If a component is non positive, it means that the actual training data is such that

- the observations do not satisfy the axiomatic properties of similarities
- the component has a strong negative correlations with some other components in the feature vector, so it is most likely encoding random redundant information emerging from the sampling problems, rather than genuine discriminatory information. Reflecting this information in the learned solution does help to get a better performance on the evaluation set where it is used as a dissimilarity. However, this does not extend to the test set.

When LDA projection vector components have all the same sign, the similarity scores are re-enforcing each other and compensating for within class variations. But for a negative component in the projection vector a positive similarity information in that dimension is not helping to get a general solution, and it is very likely that it is being used to overfit the LDA training data.

LDA is not an obvious choice for feature selection, but in the two class case of combining similarity evidence it appears that the method offers an instrument for identifying dimensions which have an undesirable effect on fusion. By eliminating every feature with a negative projection coefficient, we obtain a lower dimensional LDA projection vector with all projection coefficients positive. This projection vector is not using many of the original similarity features, and therefore performs the role of an LDA-based feature selection algorithm.

2.5.4 Results on XM2VTS of the LDA-based feature selection approach

Our experiments using this approach were conducted using the XM2VTS database [2], according to the Lausanne protocol [14] in both configurations.

For verification experiments this database was divided in three different sets: training set, evaluation set (used to tune the algorithms) and test set. We have 3 different images for every client training in Configuration I of the Lausanne protocol and 4 images for every client training in Configuration II.

An important consideration about the two different configurations is that Configuration I is using the same sessions to train and tune the algorithms, so the client attempts are more correlated than in Configuration II, where the sessions used to train the algorithms are different than those used to tune the algorithms. This means that Configuration I is likely to lead to an intrinsically poorer general solution.

In tables 3 and 4 we show the single decision stage performance with and without the LDA-based feature selection. If we compare the results in both tables we can clearly draw two main conclusions:

- The TER is lower using the LDA-based feature selection for both MLP and LDA decision fusion functions in both configurations in the test set but higher in the evaluation set.
- The difference between the FAR and FRR in the test set performance is lower for both configurations and decision fusion functions.

These two suggest that the LDA-based feature selection has enabled us to construct a solution exhibiting better generalisation properties than the one obtained when using all the features together. The stability of the operating point is also better.

On the other hand, in tables 5, 6 and 7 we have the overall system performance with and without the LDA-based feature selection algorithm. If we compare the results in tables 5 and 6, where the decision fusion function is LDA (without and with the feature selection respectively) we obtain a degradation of 5.42% in TER when using the feature selection

		Configuration I		Configuration II	
		FAR(%)	FRR(%)	FAR(%)	FRR(%)
LDA	Ev. Set	3.83	3.83	3.20	3.19
	Ts. Set	7.13	4.42	5.79	5.63
MLP	Ev. Set	0.90	0.94	0.76	0.75
	Ts. Set	2.21	7.42	2.50	9.50

Table 3: Single template performance with global thresholding and without feature selection

		Configuration I		Configuration II	
		FAR(%)	FRR(%)	FAR(%)	FRR(%)
LDA	Ev. Set	4.39	4.39	3.87	3.87
	Ts. Set	6.79	4.67	5.44	5.44
MLP	Ev. Set	2.89	2.89	2.15	2.19
	Ts. Set	4.24	5.00	3.18	6.63

Table 4: Single template performance with LDA-based feature selection and global thresholding

in Configuration I and an improvement of 6.71% in TER when using feature selection in Configuration II.

However, if we use the MLP as the decision fusion function trained with the LDA-based feature selection features, as we can see in table 7, the results in Configuration I are much better. If we do not use feature selection prior to the MLP based similarity score fusion, the results (not listed in this paper) are much worse than those listed in table 7 for both configurations, as could be expected from the highly unbalanced results shown in table 3 for the MLP fusion method.

The overall results in Configuration I should not be considered as a reflection of the generalization power of our fusion algorithms, as the poor generalization behavior is intrinsically imposed by the test protocol. Therefore it is reasonable to argue that the LDA-based feature selection allow us to improve the overall system performance.

Finally, the LDA-based selected features for both configurations can be seen super imposed over the face of one of the subjects of the database (for illustration purposes) in figure

		Configuration I		Configuration II	
		FAR(%)	FRR(%)	FAR(%)	FRR(%)
LDA	Ev. Set	1.48	1.43	0.75	0.75
	Ts. Set	3.39	3.25	1.92	2.25
MLP	Ev. Set	1.36	1.33	0.50	0.50
	Ts. Set	3.30	2.75	1.26	3.25

Table 5: Multiple template performance using LDA without feature selection for similarity score fusion, LDA and MLP as decision fusion functions and client specific thresholding

		Configuration I		Configuration II	
		FAR(%)	FRR(%)	FAR(%)	FRR(%)
LDA	Ev. Set	1.66	1.67	0.75	0.75
	Ts. Set	3.75	3.25	1.89	2.00
MLP	Ev. Set	1.83	1.83	0.50	0.50
	Ts. Set	4.65	3.00	1.05	2.75

Table 6: Multiple template performance using LDA with feature selection for similarity score fusion, LDA and MLP as decision fusion functions, and client specific thresholding

		Configuration I		Configuration II	
		FAR(%)	FRR(%)	FAR(%)	FRR(%)
LDA	Ev. Set	1.22	1.17	0.61	0.50
	Ts. Set	2.37	2.25	1.07	5.00
MLP	Ev. Set	1.11	1.00	0.52	0.50
	Ts. Set	2.20	2.25	0.93	8.00

Table 7: Multiple template performance using LDA based feature selection, MLP as similarity score fusion function, LDA and MLP as decision fusion functions and client specific thresholding

8. Note that the number and location of the selected features (40 in the configuration I and 44 in the configuration II) are very similar in both configurations, and even the values (represented in the figure by the window brightness) of the coefficients are also very similar. The stability and consistency of the features identified by the proposed algorithm is very encouraging. Moreover, the number of selected features is small enough to allow a high reduction in the computational complexity in the verification phase, and hence an important reduction (nearly a 60%) in the verification time.

2.6 Accuracy-based node selection

In the previous section, we have seen an approach to select nodes from a rectangular grid based on a Linear Discriminant Analysis. This kind of analysis is possible due to the fact that a given node represents the same facial region in every image. When locating points through Ridges and Valleys sampling, we can not assume this, so we should use another

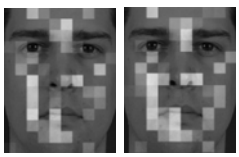


Figure 8: LDA-based selected features for configuration I (left) and configuration II (right). The brightness is proportional to the LDA projection vector coefficient

method in order to select the most discriminative nodes.

The problem can be formulated as follows: given a training image for client C, say I_{train}, a set of images belonging to the same client $\left\{I_j^c\right\}$and a set of impostor images $\left\{I_j^{im}\right\}$, we want to find which subset, $\hat{\mathcal{P}} \subset \mathcal{P}_{train}$, is the most discriminative. As long as each point $\vec{p_i}$ from $\mathcal{P}_{train}$ has a correspondent node in every other image (client or impostor, say I_{test}), we measure the individual classification accuracy of its associated jet $\mathcal{J}_{\vec{p_i}}$, and select the locations which achieve the best authentication rates, i.e., the ones with a Total Error Rate (TER) below a threshold τ. Finally, only a subset of points, $\hat{\mathcal{P}}$, is chosen per image, and the score between I_{train} and I_{test} is given by:

$$\mathcal{S} = f_{\hat{n}} \left\{ < \mathcal{J}_{\vec{p}_i}, \mathcal{J}_{\vec{q}_{\xi(i)}} > \right\}_{\vec{p}_i \in \hat{\mathcal{P}}} \tag{22}$$

This method can be applied also to select nodes (and their respective features) from a rectangular grid.

2.6.1 XM2VTS results of the accuracy-based selection approach

We performed experiments over the XM2VTS database following the Lausanne protocol on both configurations I and II. The accuracy-based selection approach was applied to:

- Rectangular grid nodes
- Ridges and Valleys nodes

The Total Error Rates over the test set are presented in table 8

	Configuration I	Configuration II
	TER(%)	TER(%)
Rectangular	4.93	2.25
Ridges	**3.61**	**2.09**

Table 8: Accuracy-based selection results for the Ridges and Valleys sampling and the rectangular grid.

3 Speaker Recognition approach

The speaker recognition system we use in our multi-biometric authentication system is a text independent speaker verifier based on GMM [16]. The acoustic parameters that we use are the Mel Frequency Cepstrum Coefficients (MFCC), their Delta and Acceleration and the Energy. The acoustic front end details can be found in [17] and [18]. A voice activity detector (VAD) based on energy is used to keep the voice frames.

The verification system is based on the likelihood ratio detection implemented by means of the GMM-UBM approach described in [19]. The verification problem is addressed as a hypothesis test between:

protocol	gender	group	optimal WP	true WP FAR	FRR	TER
		1	9,2	14,7	3,4	18,1
	f	2	10,6	5,1	14,5	19,6
		1	9,5	11,5	7,7	19,2
P	m	2	6,6	3,2	8,5	11,7
		1	2,2	3,8	0,0	3,8
	f	2	2,2	0,0	7,7	7,7
		1	0,0	0,0	17,9	17,9
MC	m	2	2,2	3,8	0,0	3,8
		1	2,6	3,2	1,7	4,9
	f	2	2,6	2,6	3,4	6,0
		1	5,1	2,6	6,8	9,4
G	m	2	1,8	3,8	1,7	5,5

Figure 9: Speaker verifier system results on BANCA with the MC, G and P experimental protocol

- H_0: The speech utterance Y comes from the speaker S.
- H_1: The speech utterance Y does not come from the speaker S.

If we know the likelihood functions $P(Y|H_i)$ then the optimal hypothesis test is:

$$\frac{P(Y|H_0)}{P(Y|H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{do not accept } H_0 \end{cases} \tag{23}$$

In this approach the hypothesis H_0 is mathematically represented by a user specific gaussian mixture model, denoted by M_S, while the alternative hypothesis is represented by another gaussian mixture model denoted as Universal Background Model (UBM), denoted as M_U. The UBM is a gaussian mixture model trained using voice segments from many different speakers. The GMM is initialized using the LBG algorithm and the training is performed using the EM algorithm. The user models are obtained adapting the UBM to the speaker's voice by means of the MAP adaptation. The logarithmic form of the hypothesis test is used due to numerical reasons:

$$\log P(Y|H_0) - \log P(Y|H_1) \geq \log \theta \Leftrightarrow \text{accept } H_0 \tag{24}$$

3.1 Results on BANCA DataBase

We have evaluated our speaker recognition system on the BANCA database with several configurations and all the protocols. We have used only the BANCA universal background model files to train the UBMs used in these experiments. Here we show the results on the protocol MC, P and G.

4 A framework for distributed audiovisual biometric authentication

In this section we summarize the main issues involved in the development of a framework for distributed biometric authentication. The algorithmic core of the framework is based

on the face and voice verification algorithms explained in sections 2 and 3, but the whole framework has been developed to easily accommodate other biometric traits using BioAPI compliant capturing devices and integrating the corresponding verifiers.

4.1 Models for distributed biometric authentication

Any biometric recognition procedure can be divided into a number of stages:

1. Acquisition of a biometric sample (device-dependent processing)
2. Extraction of a biometric template (signal processing: pre-processing, feature extraction and user-template creation)
3. Biometric template matching (pattern recognition processing).

The third of these stages usually requires prior training of user models and thresholds set up in an enrollment process whenever a new user is added to the system. The matching process may be based on identification (the biometric template is matched against all the user templates with authorization to access the system) or verification (the biometric template is matched only against the claimed user's stored biometric template). Verification is the most common matching mode for restricted access applications and corresponds to the wider security concept of identity authentication.

When dealing with distributed biometric authentication in client-server architectures, the three processes described above will provide different configurations depending on where the processes are executed. Although it is obvious that the acquisition process must be executed on the client side, there are three options remaining for extracting biometric templates and matching user templates: both these processes are performed in the client machine (pull model); the biometric template is extracted on the client side, sent to the server and matched there against the user templates (push model); or both processes are performed in the server (a variant of the push model). The pros and cons of these three possible configurations are as follows:

1. Authentication on the client side is very inconvenient. Computationally demanding operations might not be possible in the client machine, and identification mode is not feasible for medium- to large-size clients. There is also a severe security handicap, as performing all the network authentication processes on the client side is much more prone to tampering due to unsecured client machines. Note that verification requires the transaction of the claimant template from the server database to the client machine over a secure connection to avoid hacker interception. If privacy really matters, a smart-card can be used to store user biometric templates, thereby avoiding holding biometric data in a centralized server and sending it through a network. This solution is, nowadays, a very promising technological field because it combines the three authentication premises: what the user knows, possesses or is [20].
2. Biometric template extraction on the client side and template matching on the server side have the drawback of placing most of the computational burden on the client

side (pre-processing and feature extraction usually loads CPU and memory more than the matching process) and these processes are executed in a non-secure machine before the extracted biometric template is sent through a secure connection. Moreover, the authentication process itself is performed by the server, which can be properly protected.

3. Both biometric template extraction and authentication on the server side have the advantage of placing the computational load on the server side, where they can be run on powerful computers. Consequently no biometric template is sent through the network, although the acquired biometric sample is. Even though encryption is also needed for this transaction, it is important to note that some biometric samples do not constitute secret information (the face and voice of a client can be easily recorded, fingerprints are left on many objects and can be recovered, signatures can be easily photographed, etc.). Therefore, this data is less dangerous in a hacker's hands than a biometric template ready to be used in an authentication system. Finally, from the versatility and security point of view a secure server is a better configuration, as it places the bulk of the system on the server side and leaves the client side - the weakest point in the security chain - with the sole responsibility of acquiring the biometric samples. The disadvantage of this configuration is related to a wider bandwidth for the client-server connection.

The framework we have developed is based on the third of the above configurations. We define a Biometric Client Application in charge of multi-biometric sample acquisition, encryption and secure transaction; a Biometric Authentication Module with a Central Authentication Service on the server side that is in charge of extracting the biometric template, and matching and checking access privileges. The design is based on existing biometric standards - such as BioAPI and XCBF - so as to ensure interoperability and security. Details of this framework are out of the scope of this paper and can be read from [4].

4.2 Success of a distributed biometric authentication system

The performance of different biometric authentication systems reported in the scientific literature are very data-dependent and so are only really meaningful for specific tasks, specific populations and a specific acquisition set-up. Successful deployment of a distributed biometric system needs improvements in relation to certain technological and social issues. Referring merely to technological issues, there is much work to be done in regard to robustness against concurrent variability sources in the acquisition process, such as: i)different third party capturing devices even for the same biometric feature; ii)uncontrolled remote scenarios (noise, illumination, device configuration, user approach to biometrics or even technology, etc.); and iii)effects of ageing or wealth.

Even when some of the large reference datasets have quite realistic acquisition conditions, none of them has been designed to represent all the variability of a web-based large-scale application. In this sense, results of biometric verification algorithms over publicly available databases, such those shown in the previous sections of this paper, have to be understood on the limitations of the database samples.

Using a framework described in [4] it is possible to distribute the capturing process over the web and to build a dataset very useful for testing state-of-the-art biometric algorithms. The acquisition tool has been developed for audiovisual features, more challenging than other biometric features because intra-class and inter-class variability conditions are greater due to: i)a huge variety of deployed webcams (quality, resolution, driver features, built-in microphone, focal length, etc.), compared to the small set of deployed fingerprint, palmprint, iris, retina or signature acquisition devices; ii) a huge variety of acquisition scenarios: acquisition and background noise (audio and image noise), illumination, distance to the webcam and microphone, head pose, emotion and expression changes, accent and language, etc. Other biometric features can only be registered under more constrained scenarios, usually imposed by the acquisition device. And, finally, ii) greater variability in these biometric features over time (throat illness, beard, glasses, hair, ageing, etc).

5 Conclusions and future work

The main conclusions of this paper are related to the advances on face verification algorithms using local matching approaches. Results over the XM2VTS and BANCA database show that a discriminative selection or weighting of local similarities of texture information yield higher improvement of correct classification than including shape distortion information. The tests on LDA-based fusion using a global threshold show very promising results if compared to the best results of all the tests, obtained using accuracy-based node selection with user-specific thresholds. In general, a global threshold is preferred to user specific thresholds because the system will be less database-dependent and performance should not decrease too much on actual running-time. We have not performed rigorous tests on audiovisual performance using the distributed system presented in this paper. Most of the times we just combine hard decisions using logical operators or switch off one of the verifiers in bad illumination conditions or noisy environments.

The future lines of research are related to increase robustness against realistic capturing conditions. We plan to use our framework for capturing a realistic audiovisual database for desktop-based internet secure access. We are developing pose-correction algorithms for collaborative environments (no dramatic profiles shown to the webcam) and also developing fusion techniques for mixing voice and face soft decisions.

References

[1] Arun Ross, Anil K. Jain. “Multimodal Biometrics: An Overview.” In Proc. of 12th European Signal Processing Conference (EUSIPCO), pp. 1221-1224, Vienna (Austria), September 2004

[2] The extended xm2vts database. http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/

[3] The BANCA Database. http://www.ee.surrey.ac.uk/banca.

[4] José Luis Alba-Castro, Enrique Otero-Muras, Elisardo González-Agulla, Carmen García-Mateo, Oscar W. Márquez-Flórez: “An open framework for distributed biometric authentication in a web environment”, submitted to Annals of Telecommunications.

[5] Wiskott, L., Fellous, J.M., Kruger, N., von der Malsburg, C. "Face recognition by Elastic Bunch Graph Matching." IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 775-779, 1997

[6] González-Jiménez, D., Alba-Castro J.L., "Frontal Face Authentication through Creaseness-driven Gabor Jets," in Proceedings ICIAR 2004 (part II), pp. 660-667, Porto (Portugal), September/October 2004.

[7] Belongie, S., Malik, J., Puzicha J. "Shape Matching and Object Recognition Using Shape Contexts." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 24, April 2002

[8] Pujol, A., López, A., Alba, José L. and Villanueva, J.J. "Ridges, Valleys and Hausdorff Based Similarity Measures for Face Description and Matching." Proc. International Workshop on Pattern Recognition and Information Systems, pp. 80-90. Setubal (Portugal), July 2001

[9] Pearson, D.E., Hanna, E. and Martinez, K., "Computer-generated cartoons, " Images and Understanding, 46-60. Cambridge University Press, 1990

[10] López, A. M., Lumbreras, F., Serrat, J., and Villanueva, J. J., "Evaluation of Methods for Ridge and Valley Detection," IEEE Trans. on PAMI, 21(4), 327-335, 1999

[11] Huttenlocher, D.P. et al.,"Comparing images using the hausdorff distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 3, pp. 850-863, 1993

[12] Dubuisson, M.P. and Jain, A.K.,"A modified hausdorff distance for object matching," in Proceedings IEEE International Conference on CVPR, 1995

[13] Paglieroni, D., "Distance transforms: Properties and machine vision applications," CVGIP:Graphical models and image processing, vol. 54, no. 1, pp. 56-74, 1992

[14] Luttin, J. and Maître, G., "Evaluation protocol for the extended M2VTS database (XM2VTSDB)." Technical report RR-21, IDIAP, 1998.

[15] Enrique Argones-Rúa, Josef Kittler, José Luis Alba-Castro, Daniel González Jiménez, "Information fusion for local Gabor features based frontal face verification". Accepted in International Conference on Biometrics, Hong-Kong, January 5-7, 2006

[16] Leandro Rodríguez-Liñares, Carmen García-Mateo, José Luis Alba-Castro: "On Combining Classifiers for speaker authentication," Pattern Recognition, vol. 36, pp.347-359. 2003.

[17] ES 202 212 V1.1.2 (2005) : Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm.

[18] Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland: The HTK Book (for HTK Version 3.2.1), 2002.

[19] Douglas A. Reynolds, Thomas F. Quatieri, Robert B. Dunn: "Speaker verification using adapted gaussian mixture models," Digital Signal Processing 10, 19-41, 2000.

[20] Menkus, B. "Understanding the Use of Passwords," Computers and Security 7(2), 132-136 (1988).

Error correcting codes embedding of mutual information trees *

Oriol Pujol†, Petia Radeva‡, Jordi Vitrià‡
† Dept. de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona,
Gran Via 585, Barcelona, 08007, Spain.
E-mail: oriol@maia.ub.es
‡Computer Vision Center and Dept. Ciencies de la Computacio,
Edifici O, Campus UAB, Bellaterra, 08193, Spain.
E-mail: petia@cvc.uab.es, jordi@cvc.uab.es

Abstract

We present a heuristic method for learning error correcting output codes matrices based on a hierarchical partition of the class space that maximizes a discriminative criterion. To achieve this goal the optimal codeword separation is sacrificed in favor of a maximum class discrimination in the partitions. The creation of the hierarchical partition set is performed using a binary tree. As a result, a compact matrix with high discrimination power is obtained. Our method is validated using the UCI database, and applied to a real problem, the classification of traffic sign images.

Keywords: Multiple classifiers, Multi-class classification, Visual Object Recognition.

1 Introduction

The task of supervised machine learning can be seen as the problem of finding an unknown function $C(x)$ given the training pair set of examples $< \mathbf{x}_i, C(\mathbf{x}_i) >$. $C(x)$ is usually a set of discrete labels. For example, in face detection $C(x)$ is a binary function $C(x) \in \{\text{face}, \text{non-face}\}$, in optical digit recognition $C(x) \in \{0, \ldots, 9\}$.

In order to address the binary classification many techniques and algorithms have been proposed: decision trees, neural networks, large margin classification techniques , etc. Some of those methods can be easily extended to multiclass problems. However, some other powerful and popular classifiers, such as AdaBoost [3] and Support Vector machines , do not extend to multiclass easily. In those situations, the usual way to proceed is to reduce the complexity of the multiclass problem into multiple simpler binary classification problems.

*This work was supported by FIS: PI031488, and FIS network: G03/185 of MEC.

There are multiple approaches for reducing multiclass to binary classification problems. The simplest approach considers the comparison between each class against all the others. This produces N_c binary problems, where N_c is the number of classes. Other researchers suggested the comparison of all possible pairs of classes [4], resulting in a $N_c(N_c-1)/2$ set of binary problems. Dietterich et al. [6] presented a general framework in which classes are classified according to a set of binary error correcting output codes (ECOC). In this approach the problem is divided in n binary classification subproblems, where n is the error correcting output code length $n \in \{N_c, \ldots, \infty\}$. The output of all classifiers must be then combined (traditionally using Hamming distance). Dietterich approach was improved by Allwein et al. [5] by introducing an uncertain value in the ECOC design and exploring alternatives for combining the resulting outputs of the classifiers. In particular, they introduced loss-based decoding as a way of combining the classifiers. Recently, Passerini et al [2] proposed a new decoding function that combines the margins through an estimate of the class conditional probabilities.

Though most of the improvements in error correcting output codes have been made in the decoding process, little attention has been paid to the design of the codes themselves. Crammer et al. in [1] were the first authors to report improvements in the design of the codes. However, the results were rather pessimistic since they proved that the problem of finding the optimal discrete codes is computationally intractable since it is NP-complete.

It is our purpose in this paper to reopen the problem of discrete ECOC design by proposing an heuristic method that not simply gives an efficient and effective method for ECOC design but leads to compact codes of $N_c - 1$ bits (binary problems).

The method we propose renders each column of the output code matrix to a problem of finding the binary partition that divides the whole set of classes so that the discriminability between both sets is maximum. The criterion used for achieving this goal is based on the mutual information between the data of each set and its class label. Since the problem is defined as a discrete optimization process, we propose to use floating search methods as sub-optimal search procedures for finding the partition that maximizes the mutual information. The whole ECOC matrix is created with the aid of an intermediate step formulated as a binary tree. With this formulation we ensure that we decompose the multiclass problem into $N_c - 1$ binary problems.

The paper is divided in the following sections: section 2 provides a brief introduction to error correcting output codes, section 3 describes the discriminant ECOC technique as well as the theory of the methods involved in its creation. Section 4 shows empirical results of the proposed method and section 5 concludes the paper.

2 Error correcting output codes

Error correcting output codes were born as a general framework for handling multi-class problems [6]. The basis of this framework is to create a codeword for each class (up to N_c codewords). Arranging the codewords as rows of a matrix they define the "coding matrix" M, where $M \in \{-1, 1\}^{N_c \times n}$, and n is the code length.

From the point of view of learning, the matrix M is interpreted as n binary learning problems, one for each column. Each column defines a partition of classes (coded by +1,-1 according to their class membership). As a result of the outputs of the n binary classifiers a code is obtained for each data point in the test set. This code is compared with the base codewords of each class defined in the matrix M, and the data point is assigned to the class with the "closest" codeword.

A generalization of this process is provided in [5]. The main difference in terms of the coding matrix is that it is taken from a larger set $M \in \{-1, 0, 1\}^{N_c \times n}$. In this approach some entries in the matrix M can be zero indicating that a particular class is not significative for a given classifier. In practical applications this means that the classifier omits all examples for which $M = 0$.

Table 1: Example of the M matrices for a 4-class problem. (a) 1-against-all matrix (b) all-pairs matrix.

	h1	h2	h3	h4
C1	+1	-1	-1	-1
C2	-1	+1	-1	-1
C3	-1	-1	+1	-1
C4	-1	-1	-1	+1

(a)

	h1	h2	h3	h4	h5	h6
C1	+1	+1	+1	0	0	0
C2	-1	0	0	+1	+1	0
C3	0	-1	0	-1	0	+1
C4	0	0	-1	0	-1	-1

(b)

Table 1 provides two examples of M matrices applied to a four class problem. C_i is the class label and h_i is the binary classifier (hypothesis). In the case of one-against-all classification, M is a $N_c \times N_c$ matrix in which all diagonal elements are set to +1 while the rest are set to -1. In the case of all-pairs classifiers, M is a $N_c \times N_c(N_c - 1)/2$ matrix in which each column is set to zero except for a given pair. One of the pair elements is set to 1 and the other to -1.

Although both strategies (one-against-all and all-pairs) address the problem of multi-class from binary problem, Allwein et al. showed in [5] that all-pairs outperformed the one-against-all strategy. However, the complexity of all-pairs is superior to the one-against-all one.

Several other heuristics for creating ECOC matrices are proposed in [6] such as exhaustive codes, sparse matrix coding and compact matrix coding. All those codes

are defined independent of the data set to be classified satisfying two properties:

- **Row separation.** Each codeword should be well-separated in Hamming distance from each of the other codewords.
- **Column separation.** Each column h_i should be uncorrelated with all the other columns $h_j, j \neq i$. This property is achieved if the Hamming distance between columns is large. The largest distance is obtained when compared with the complement of each column.

As we mentioned before, the codeword resulting of applying the different hypotheses to a given instance x should be combined. If we denote $f(x) = (f_1(x), \ldots, f_n(x))$ the vector of predictions for the sample x, the combination of the n outputs assigns one of the N_c labels. The simplest way of decoding a vector $f(x)$ is the *Hamming decoding*. This method looks for the minimum distance $d_H(M(r,.), f(x))$ between the prediction and the codewords:

$$\hat{y} = \underset{r}{argmin} \quad \big(d_H(M(r,.), f(x))\big), \quad d_H(M(r,.), f(x)) = \sum_{s=1}^{n} \big(\frac{1 - \text{sign}(M(r,s) f_s(x))}{2}\big)$$

where sign(z) is $+1$ if $z > 0$, -1 if $z < 0$ and 0 otherwise. $M(r,.)$ designates the codeword r in the matrix and $\hat{y} \in \{1, \ldots, N_c\}$ is the predicted label.

The next generalization made in ECOC comes from Crammer et al. [1]. In their work, they change the discrete values of the ECOC matrix for continuous ones. Thanks to this change they are able to find a method for creating application dependent ECOC matrices and solve an, otherwise, NP-complete problem.

Our work reopens the problem of the design of the discrete coding matrix. The main difference between our work and the rest is that while the rest of the discrete approaches ignores completely the structure of the given problem, we create the coding matrix according to the particularities of the data we are dealing with. In order to achieve our goal we must relax the conditions of row and column separation. We trade the optimality in the codewords for maximum class separation in the partitions.

3 Discriminant ECOC

Discriminant ECOC is born as a result of three processes: first, a heuristic for the design of the ECOC matrix; second, the search for high performance classification using the minimum number of classifiers; and third, a tool to describe the classification domain in terms of class dependencies. We have seen in the previous section

Table 2: **CCBT algorithm**

[Initialization:]

Root node : N_0
N_0.set $= \{C_i \mid \forall i \in \{1, \ldots, N_c\}\}$.
Create a list of nodes $L_N = \{N_0\}$

Step 1. $N_t = \text{first}(L_N)$ Remove the first node of L_N

Step 2. Use *floating search* in conjunction with *fast quadratic mutual information* to find the most discriminant partition $S_j = \{C_i\} \subset N_t.\text{set}, j \in \{1, 2\}$.

Step 3. Create a node for each partition set $< N^1, N^2 >$ and fill the field "set" with the partition set labels, S_j.

Step 4. Add those nodes to the list of nodes $L_N = L_N \cup \{N^1, N^2\}$ if length(N^k.set) > 1

Step 5. Go to step 1 if there are still nodes in the list.

that one-against-all and all-pairs classification strategies are the classic examples for the binary and ternary valued ECOC design, respectively. Our approach relaxes the strong assumption of the one-against-all classification approach by allowing the classes to organize in maximally discriminant sets while keeping the number of classifiers low. On the other hand, the all-pairs approach considers each class unrelated to the rest. It is our desire to exploit the inner dependencies among classes in the classification domain. As a result of these specifications, we define the discriminant ECOC.

3.1 Design of the Discriminant ECOC

The goal of this work is to find a compact multiclass (in terms of codeword lenght) codeword matrix M with high discriminative power. To achieve this goal, we will use as an intermediate step a *Column Code Binary Tree* (CCBT) where each node of the tree defines a partition of the classes and therefore a column of the matrix M. The partition at each node must satisfy the condition to be highly separable in terms of discrimination. To achieve this goal the partition obtained is the result of the maximization of the quadratic mutual information between the data and the

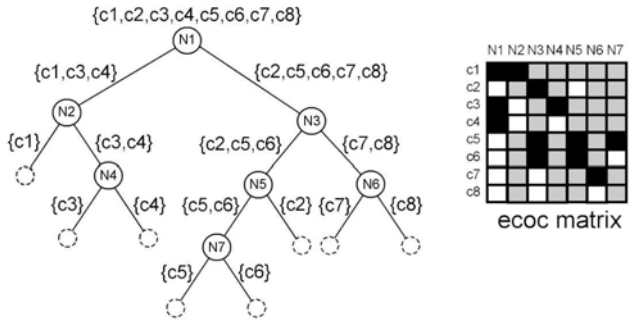


Figure 1: Example of conversion from the binary tree to the ECOC matrix.

labels of the partition. The algorithm used for the maximization is the floating search method, that will be introduced in the next subsection.

Therefore, the general algorithm to find the matrix M is as follows,

General steps

- Create the *Column Code Binary Tree*
  - Recursively, find the most discriminant partition of the parent node (N_i) class set using *floating search* with *fast quadratic mutual information* criterion.
- Assign to the column i of matrix M the code of node N_i of the tree:

$$M(:, i) = N_i.\text{code}$$

Table 2 details a possible algorithm for creating the CCBT. The tree is a mean to find the codewords. The final matrix M is composed by the codes obtained at each node (except for the leaves). Those codes would be placed as columns in the coding matrix ($M(., i)$). To create each column code we use the relationship between a node and its child nodes. The rules to create the column code are the following:

- All the elements in the column code $M(., i)$ related to a class C_r not appearing in the set of classes of the node are set to zero. $M(r, i) = 0 \quad \text{if} \quad C_r \notin N_i.\text{set}$
- The elements in the column code of the node related to the classes of one of the two child nodes of the given node are set to $+1$. $M(r, i) = +1 \quad \text{if} \quad C_r \in N_i^1.\text{set}$
- The remaining elements are set to -1. $M(r, i) = -1 \quad \text{if} \quad C_r \in N_i^2.\text{set}$

Note that the number of n columns coincides with the number of the intermediate nodes. It is easy to see that in any binary tree the number of intermediate nodes is $N_c - 1$ given that the number of leaves is N_c. Therefore, by means of the CCBT we can assure that the codeword will have length $N_c - 1$.

Figure 1 shows an example of a CCBT for 8 classes. On the right side of the figure, we show the resulting discriminant ECOC matrix. The white squares are +1, black squares are -1 and gray squares have 0 value. Observe, for instance, that column $N5$ corresponds to the partition $\{c5, c6\}$ and $\{c2\}$. On the other hand, if we look at the rows of the matrix, the codeword associated to class 6 ($c6$) is $\{1, 0, -1, 0, -1, 0, 1\}$.

As a result of this process the discriminant ECOC matrix is created. From a more general point of view, the creation of the ECOC matrix is one of the parts involved in the multiclass classification technique. The other two remaining parts to be defined are the classification technique and the decoding strategy. In this paper we have chosen AdaBoost [3] as a classification technique, since it is becoming a state-of-the-art binary classification technique (that has inherent problems to extend to multiclass classification). The chosen decoding metric is the Euclidean distance to the codewords. We have seen that Euclidean and Hamming distance have the same performance for classic M matrices. However, our method does not necessarily fulfill the row and column separation properties described in the former section. This is due to the fact that similar classes are translated into closer codewords. In this scenario, Hamming distance is not well suited to handle those variations and alternative decoding metrics have to be used.

Recalling the algorithm described in table 2, a maximization process is needed to obtain the partition of the classes in two sets. Although looking for the best partition set requires of an exhaustive search among all possible partitions, due to the impracticability of this endeavor a suboptimal strategy must be used. The strategy chosen is the *floating search method*. The following subsection details this method that allows the problem to be computationally feasible.

3.2 Floating Search Methods

The *Floating search method* [8] was born as a suboptimal search method for alleviating the prohibitive computation cost of exhaustive search methods in feature selection. This method lies in the family of sequential search methods where one of the most favored search procedures for its effectiveness is the *plus-l take away-r* method.

The heuristic basis and the main constraint of most sequential methods are that the search criterion has to be monotonic. This implies that when adding a new item

the search criterion to be maximized does not decrease. However, this condition does not always hold in many practical cases. In particular, researchers present several partially successful approaches to cope with this problem using Montecarlo approaches or genetic algorithms [7].

Pudil et al. introduced in [8] a family of suboptimal search algorithms called *floating search methods* that resulted effective in high dimensional problems. Furthermore, these methods allowed the search criteria to be non-monotonic, thus solving the main constraint of many sequential methods. This family of methods is directly related to the *plus-l take away-r* algorithm. However, the first approach differs from *plus-l take away-r* algorithm in the fact that the number of forward and backtracking steps are not decided beforehand.

Floating search methods can be described as a dynamically changing number of forward steps and backward steps as long as the resulting subsets are better than the previously evaluated ones at that level. In this sense this method avoids nesting effects that are typical of sequential forward and backward selection while equally being step-optimal since the best (worst) item is always added (discarded) to (from) the set. Since backtracking is controlled dynamically, no parameter setting is needed.

The algorithm presented in table 3 describes the top-down approach which is called Sequential Forward Floating Search (SFFS) algorithm. This one begins with an empty set X_0 and is filled while the search criterion applied to the new set increases. The most significant item with respect to X_k is added at each inclusion step. In the conditional exclusion step, the worst item is removed if the criterion keeps increasing. In our case Y is the set of classes to be partitioned.

In our approach, the criterion used for designing this partition is related to the discriminability between the class sets. We use mutual information to that effect. Our goal is to maximize the mutual information between the data in the sets and the class labels of the partitions.

3.3 Fast Quadratic Mutual Information

Mutual information (MI) is a well known criterion to compute the amount of information that one random variable tells about another one. In classification theory, this measure has been shown to be optimal in terms of class separation [11] [10], allowing to take into account high-order statistics. MI also bounds the optimal Bayes error rate. However, mutual information is not widely used due to the difficulties derived from its computation.

Though evaluating MI in low dimensional spaces (small number of random variables) can be feasible through histograms, it can not be easily accomplished in high dimensional ones due to sparsity of data. However, Principe et al. [10] presented a

Table 3: **SFFS Algorithm**

Input:

$Y = \{y_j | j = 1..D\}$//Available items//

Output:

$X_k = \{x_j | j = 1..k, x_j \in Y\}, \quad k = 0, 1, ..D$

Initialization:

$X_0 = \{\emptyset\}; \quad k = 0$

Termination:

Stop when the criterion does not increase $J(X_k) \approx J(X_{k-1})$

Step 1 (Inclusion)

$x^+ = \underset{x \in Y - X_k}{argmax}\ J(X_k \cup x)$
$X_{k+1} = X_k \cup x^+, \quad k = k + 1$

Step 2 (Conditional exclusion)

$x^- = \underset{x \in X_k}{argmax}\ J(X_k - x)$
if $J(X_k - x^-) > J(X_{k-1})$ then
  $X_{k+1} = X_k - x^-, \quad k = k + 1$
  go to **Step 2**
else
  go to **Step 1**

feasible method for computing entropy estimators using Renyi's formulation when coupled with Parzen window density estimation. Based on this method, they heuristically obtained a measure for mutual information.

This work has been recently modified and extended by Torkkola et al. [11] by relating mutual information to divergence measures. Using this extension the authors provide the base for computing "quadratic mutual information" in a simple and fast way.

Let $\mathbf{x}$ and $\mathbf{y}$ represent two feature vector sets where $\mathbf{x} = \{x_i \in \Re^d\}$ and $\mathbf{y} = \{y_i \in \Re^d\}$ and $p(\mathbf{x})$, $p(\mathbf{y})$ are their respective probability density functions. The mutual information measures the dependence between two probability distributions and is defined as follows,

$$I(\mathbf{x},\mathbf{y}) = \int\int p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) dxdy \tag{1}$$

Observe that mutual information is zero if $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$. It is important to note that equation (1) can be seen as a Kullback-Leibler divergence,

$$K(f,g) = \int f(y) log(\frac{f(y)}{g(y)}) dy$$

where $f(y)$ is replaced with $p(x,y)$ and $g(y)$ with $p(x)p(y)$.

Alternatively, Kapur et al. argued in [9] that if our goal is to find a distribution that maximizes or minimizes the divergence, several axioms can be relaxed and the resulting divergence measure is related to $D(f,g) = \int (f(y) - g(y))^2 dy$. It was proved in [11] that maximizing $K(f,g)$ is equivalent to maximizing $D(f,g)$. Therefore

$$I_Q(\mathbf{x},\mathbf{y}) = \int\int (p(\mathbf{x},\mathbf{y}) - p(\mathbf{x})p(\mathbf{y}))^2 dxdy \tag{2}$$

The estimation of the density functions can be done using the Parzen window estimator. In that case, when combined with Gaussian functions we can use the following property: Let $N(y,\Sigma)$ be a d-dimensional gaussian kernel, it can be shown that,

$$\int N(y - a_1, \Sigma_1) N(y - a_2, \Sigma_2) = N(a_1 - a_2, \Sigma_1 + \Sigma_2)$$

Observe that the use of this property avoids the computation of one integral function. Threrefore, given N_y data points, $p(y)$ and $p(x|y)$ can be written as,

$$p(y) = \frac{1}{N_y} \sum_{i=1}^{N_y} N(y - y_i, \sigma I), \quad p(x|y) = \frac{1}{N_y} \sum_{j=1}^{N_y} N(x - y_j, \sigma^2 I)$$

Let us define the notation for the practical implementation of I_Q: Assume that we have N samples in the whole data set; J_p are the samples of each class c_p; N_C stands for the number of classes; x_l stands for the l-th feature vector of the data set and x_{pk} is the k-th feature vector of the set in class c_p. Expanding equation (2) and using a Parzen estimate with a symmetric kernel with width σ, we obtain the following equations,

$$I_Q(\mathbf{x},\mathbf{y}) = V_{IN} + V_{ALL} - 2V_{BTW}$$

where,

$$
\begin{aligned}
V_{IN} &= \int\int p(\mathbf{x},\mathbf{y})^2 dxdy = \frac{1}{N^2}\sum_{p=1}^{N_C}\sum_{l=1}^{J_p}\sum_{k=1}^{J_p} N(x_{pl}-x_{pk}, 2\sigma^2 I), \\
V_{ALL} &= \int\int p(\mathbf{x})^2 p(\mathbf{y})^2 dxdy = \frac{1}{N^2}\sum_{p=1}^{N_C}(\frac{J_p}{N})^2\sum_{l=1}^{N}\sum_{k=1}^{N} N(x_l-x_k, 2\sigma^2 I), \quad (3) \\
V_{BTW} &= \int\int p(\mathbf{x},\mathbf{y}) p(\mathbf{x}) p(\mathbf{y}) dxdy = \frac{1}{N^2}\sum_{p=1}^{N_C}\frac{J_p}{N}\sum_{l=1}^{N}\sum_{k=1}^{J_p} N(x_l-x_{pk}, 2\sigma^2 I)
\end{aligned}
$$

In practical applications, σ is set to the half of the maximum distance between samples as proposed by Torkkola in [11].

4 Experimental Results

In this section we describe and discuss the experiments we have performed with natural data from two different environments. First, we validate the approach using data from the UCI repository. Afterwards, we apply this approach to a real problem: traffic sign recognition.

4.1 Validation in UCI database

To validate our approach we begin with an analysis using the standard UCI database [12]. This database is a well-known database for evaluation and comparison of classifiers. We have chosen a very popular binary learner for these experiments: AdaBoost [3] with 40 weak learners per strong classifier (h_j). We have selected from the UCI database the following datasets: Iris, Wine, Balance-Scale, New-Thyroid, Dermatology, Glass, Ecoli, Yeast, Vowel and Abalone. The properties of the datasets are described in table 4. The experiments have been performed using a 10 fold cross-validation strategy.

We tested 3 different types of output codes: one-against-all, all-pairs and discriminant ECOC. We have decided to use only these codes because of different reasons: first, we choose all-pairs to compare because in [5], the authors showed that all-pairs is one of the most discriminant codes, better than sparse and dense code approaches. Second, we choose to compare with one-against-all because it is the only coding reported in literature [5], up to our knowledge, comparable in terms of number of classifiers needed in the multiclass classification process.

Table 4: Description of the datasets used in the experiments

Problem	#Examples	#Attributes	#Classes
iris	150	4	3
glass	214	9	7
wine	178	13	3
ecoli	336	8	8
balance-scale	625	4	3
yeast	1484	8	10
new-thyroid	215	5	3
vowel	998	10	11
dermatology	366	34	6
abalone	4177	8	28

Table 5: Mean error rate for different UCI datasets.

Data Set , Method	DECOC	1 vs 1	1 vs All
Iris	**4.29%**	4.50%	5.40%
Glass	**25.65%**	27.74%	35.96%
Wine	4.94%	**4.42%**	6.72%
Ecoli	19.15%	**17.38%**	23.30%
Balance-Scale	10.12%	8.77%	**7.95%**
Yeast	47.0%	**46.35%**	48.24%
New-Thyroid	5.23%	**3.32%**	6.23%
Vowel (general)	20.10%	**18.16%**	51.81%
Dermatology (Hamming)	11.78%	6.20%	11.94%
Dermatology	**6.76%**	6.90%	11.72%
Vowel (T-T)	60.36%	**52.59%**	76.40%
Abalone	76.01%	**74.05%**	99.37%

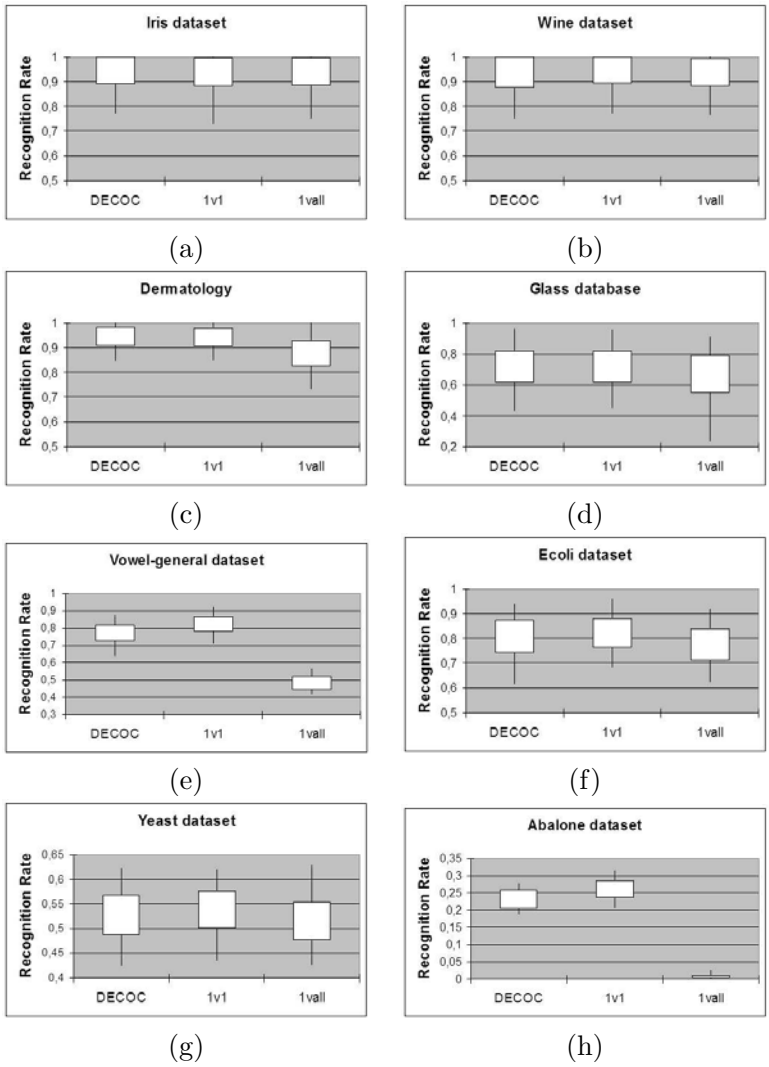


Figure 2: Comparison of the recognition rate statistical behavior among DECOC, all-pairs (1v1) and one-against-all (1vall).

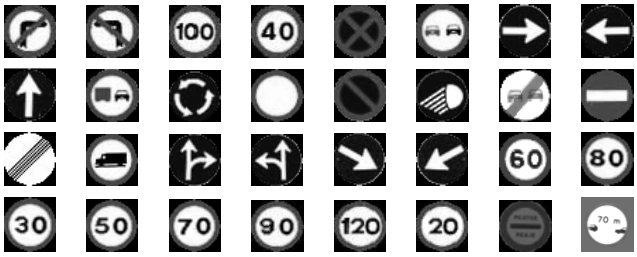


Figure 3: The 32 different signal classes to recognize.

In order to compare the methods we choose the mean error rate (displayed in table 5) and the maximum, minimum and standard deviation of the recognition rate of the classification methods (illustrated in figure 2). In the table, discriminant ECOC, one-against-all and all-pairs are abbreviated using "DECOC", "1 vs all" and "1 vs 1", respectively. We can see that our method outperforms the one-against-all approach easily, and in most of the databases is comparable to all-pairs. However, our method just needs $N_c - 1$ classifiers instead of $N_c(N_c - 1)/2$ that derives from the all-pairs approach. This is a very significant gain when the number of classes increases. For instance, in the abalone dataset (we have 28 classes), we need to train 378 classifiers in the all-pairs approach. However, we just need 27 in our approach. This allows our approach to be used in applications where time is a crucial constraint, such as on-line applications, real time or near real time applications and transductive learning, where retraining is needed.

4.2 Traffic sign recognition

The proposed approach was used in an online traffic sign detection and recognition project for guided navigation. In particular, we are concerned with the traffic sign recognition part. In this problem we have a set of 32 different signs that have to be distinguished. An example of each class is illustrated in figure 3.

We used the three different approaches to compare the performance in this problem. The binary base classifier was AdaBoost. The training set was extracted from 8 car drive records at different locations, highways and local roads. The total number of examples sums 2217. This problem has an additional difficulty since the number of samples for each class is very different. This means that we are in front of an imbalanced class problem with classes clearly under-represented. Each extracted sign image measures 35×35 pixels. The data has not been preprocessed and has been used raw data as input to our learners. We have used two different car drive records (dataset1 and dataset2, corresponding to a highway and a local road, respectively)

Table 6: Error rates for traffic sign recognition.

Test sequence	DECOC	1 vs 1	1 vs All
dataset1	**9.38%**	13.14%	27.23%
dataset2	**12.68%**	15.85%	29.17%

as test sets. The total number of traffic signs in the test records sums 600. The results obtained in our experiments are summarized in table 6.

We can observe that the behavior of the three methods follows the guidelines obtained when we validated the method using the UCI datasets. However, in this case our method improves not only the one-against-all approach but also the all-pairs. This success is reinforced by the fact that our approach uses only 31 classifiers instead of the 496 classifiers used by the all-pairs approach, thus increasing the computational efficiency of the whole process, training and test.

The improvement of our method over the all-pairs one is due to the fact that our method is able to generalize better than all-pairs in front of classes with a small number of samples. In this case all-pairs approach fails because of the imposing to find the class boundary. However, since our partitions gather together several classes, this problem does not affect.

Figure 4 shows the discriminant ECOC matrix for the signal recognition application. If one analyzes the resulting partitions created by our method, one can see that several groups make sense in terms of perceptual discrimination. Let us take for instance the partitions created to train the seventh column. As we observe all two digit speed signals have been grouped in front of three digits speed signals and also in front of signals that contain two objects that are not digits. This perceptual partition results encourage us to believe that the method is also capable of meaningful class clustering.

5 Conclusion

We have introduced a new algorithm, discriminant ECOC, for designing compact error correcting output codes. The result is a multi-class classifier that runs faster (since it uses fewer number of classifiers) and requires less training time, while maintaining (and improving in some cases) the performance of the all-pairs approach. This approach is also the first one to deal successfully with the problem of the design of application dependent discrete error correcting output code matrices.

We have applied the discriminant ECOC algorithm to the UCI database for validating purposes and to a real computer vision application: traffic sign recognition.

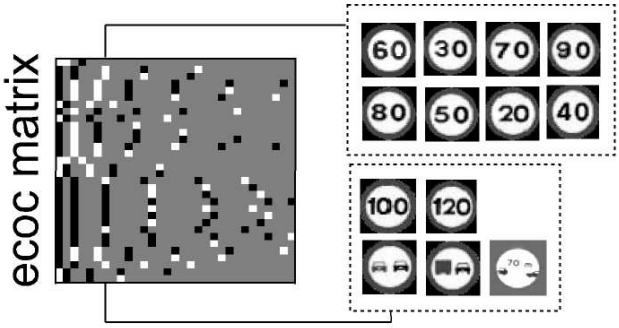


Figure 4: Discriminant ECOC matrix created for the signal recognition system and partition at the seventh column

As a result, our method compares favorable to all-pairs and clearly outperforms one-against-all methods. On the other hand, in the traffic sign recognition it outperforms the rest of the methods. We believe that discriminant ECOC algorithm reopens the problem of the design of error correcting output codes and offers a very promising research line.

References

[1] K. Crammer and Y. Singer, "On the Learnability and Design of Output Codes for Multiclass Problems," *Machine Learning,* vol. 47, no. 2-3, pp. 201-233, 2002.

[2] A. Passerini, M. Pontil and P. Frasconi, "New Results on Error Correcting Codes of Kernel Machines," *IEEE Trans. on Neural Networks,* vol. 15, no. 1, pp. 45-54, 2004.

[3] Y. Freund and R.E. Shapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[4] T. Hastie and R. Tibshirani, "Classification by Pairwise Coupling," *The Annals of Statistics*, vol. 26, no. 2, pp. 451-471, 1998.

[5] E.L Allwein, R.E Shapire and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers," *Journal of Machine Learning Research,* vol. 1, pp. 113-141, 2000.

[6] T.G. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes," *Journal of Atificial Intelligence Research*, vol. 2, pp.263-286, 1995.

[7] W. Siedlecki and J. Sklansky, "On Automatic Feature Detection," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 2, pp. 197-220, June 1988.

[8] P. Pudil, F. Ferri, J. Novovičová and J. Kittler, "Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions", *Proceedings of ICPR94*, pp.279-283, 1994.

[9] J.N. Kapur and H.K. Kesavan, *Entropy Optimization Principles with Applications*, Academic press, San Diego, London, 1992.

[10] J. Principe, D. Xu and J. Fisher III, "Information Theoretic Learning," *Unsupervised Adaptive Filtering*, New York, NY: Wiley, 2000.

[11] K. Torkkola, "Feature Extraction by Non-Parametric Mutual Information Maximization," *Journal of Machine Learning Research,* vol. 3, pp. 1415-1438, 2003.

[12] P.M. Murphy and D.W. Aha, *UCI Repository of machine learning databases,* [http://www.ics.uci.edu/ mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science. 1994.

Efficient search with tree-edit distance for melody recognition*

David Rizo, Francisco Moreno-Seco, José M. Iñesta,
and Luisa Micó
Dept. Lenguajes y Sistemas Informáticos
Universidad de Alicante, E-03071 Alicante, Spain
{drizo,paco,inesta,mico}@dlsi.ua.es

Abstract

The search of a given melody in large data-bases is one of the problems in the modern topic of music information retrieval (MIR). A huge amount of music files in symbolic formats can be found today in the Internet, and this has motivated new challenges for identification and categorization of music data. A number of pattern recognition techniques can be used to solve this problem. In this paper we explore the capabilities of trees to provide an expressive representation of music information. Trees are compared to string representations in terms of dissimilarity measures, using edit distances. The high computational cost of tree edit distances needs of complexity reduction techniques to be applied. Partial tree edit distances will be considered in order to solve this problem. Also, a new approximate nearest neighbour search for non-vector representation of patterns (such as trees) is applied to speed up the classification. The combination of both techniques produces a significant reduction in classification error rates of string representations while keeping similar classification times.

Keywords: Nearest neighbour, computer music, structural recognition, tree edit distance, complexity reduction

1 Introduction

1.1 Context and previous works

The search of a particular melody in large data-bases is a great challenge that needs of accurate and efficient recognition algorithms. In the past few years, the amount of music files available, such as MP3, MIDI, XML representations, ringtones, etc. has grown very quickly. Even different variations of the same original theme can

*This work was supported by the projects Spanish CICYT TIC2003–08496–C04, partially supported by EU ERDF, and Generalitat Valenciana GV043-541.

be found, so another difficult problem is the recognition of different interpretations of the same melody. Also, music identification from inaccurate or distorted queries is needed. The approaches to solve those problems are part of the modern topic of music information retrieval (MIR) and have lots of applications like organization and indexing digital libraries or copyright management, to name just two of them.

Some recent papers explore the capabilities of pattern recognition algorithms to recognize music data. These data can be classified into two main categories: digitized sounds and symbolic sequences. With regard to digital sounds, a number of works explore the capabilities of pattern recognition algorithms for finding different categories in music. A few of them are cited next, covering a representative range of applications.

In a recent work [1], the authors evaluate the ability of different sets of audio features, like low-level signal properties, mel-frequency spectral coefficients, and linear prediction coefficients, for classifying digital sound segments into a set of sound classes, like sung music, instrumental music, speech, noise, etc. The classification is performed through a Bayesian approach. In [2] a system based on neural networks and support vector machines is presented for classifying audio fragments into a given list of sources or artists. Also in [3] a neural system to recognize music types from sound inputs is described. Other audio classifications are based on clustering analysis. In [4], the authors use self-organizing maps (SOM) to pose the problem of organizing music digital libraries according to sound features of musical themes, in such a way that similar themes are clustered, performing a content-based classification of the sounds.

On the other hand, symbolic sequences refer to digital scores available in a number of public formats, like MIDI [5] or MusicXML [6]. Different pattern recognition techniques have been applied to process and classify these sequences. In [7], the authors show the ability of grammatical inference methods for modelling musical style. Stochastic finite automata for a number of musical styles are inferred from the training set, and then they are utilized to parse and classify new melodies into the selected styles. In [8], the authors compare the performance of different pattern recognition paradigms to recognize music style using descriptive statistics of pitches, intervals, durations, rests, etc. Other approaches like hidden Markov models [9] or classifier ensembles [10] have been used to recognize melodies, styles, authors or performers from symbolic data.

The work presented in this paper uses symbolic data as input and deals with the recognition of melodies with different degrees of distortion. Preliminary results of the proposed technique have been published in former works [11, 12].

1.2 Objectives of this paper

Some papers [13, 14] have discussed the sensitivity of the recognition algorithm performance to the encoding scheme used to represent the melodic sequences. The authors point out the need of designing an appropriate representation framework, because otherwise the algorithms can fail in their classification task. In [13], the authors present a number of string representations of melodies in terms of symbols coding the sequence of notes. The results for the different codings are compared, showing that the way in which the melody is coded strongly conditions the outcome of the string classification algorithms.

One possible alternative to music string representation are trees [15, 11, 16]. This data structure has the advantage of being able to represent music note duration implicitly, so there is no need of designing an alphabet of symbols to represent durations and time proportions. This way, tree representation of music will be less sensitive to coding. On the other hand, tree construction, processing, and analysis are more expensive than for strings.

There is a need of studying whether trees are useful for posing this sort of problems, and what has to be taken into account to do it. In this paper, a method for representing melodies as trees is presented. Also, a set of rules are introduced to label the tree nodes and reduce the initial size of the tree in order to deal with complexity.

Once the melodic sequences are tree-coded, an efficient classification algorithm is needed. The trees are compared in terms of dissimilarity measures, using tree edit distances, that are provided to a nearest neighbour (NN) search algorithm. The high computational cost of tree edit distances [17] needs of complexity reduction techniques to be applied. Two cooperative techniques are combined in order to reduce computational load, keeping the accuracy in a high standard. The first uses a tree edit distance that is cheaper than the full edit distance, and the second is based on using a new approximate NN search instead of exact NN search to reduce the number of distance computations, and thus to reduce classification times. This new approximate NN search is the extension of previous works on approximate NN search for prototypes codified as vectors to non-vector representations of prototypes. Classification error increases with approximate search, but avoids the computation of a large amount of expensive tree edit distances. The combination of both techniques reduces the cost, reaching times even better to those of string-coded representations.

Firstly, the method for tree construction will be presented and how it deals with the notation problems that may appear. Secondly, a set of rules for tree simplification is described. Thirdly, the methods for tree comparison and efficient neighbour search are explained. Finally, the results are presented and some conclusions are

stated.

2 Representations for music sequences

A melody has two main dimensions: rhythm and pitch. Basically, the first is determined by note onset times and durations, and the second by the fundamental frequencies of the notes. The main methods for melody search are based on different string codings of those dimensions [7, 14], focusing mainly on pitch. Nevertheless, rhythm is an important component of music. One can find melodies having the same sequence of note pitches but sounding completely different due to the differences in their durations.

In string representations, note durations are coded with explicit symbols, but trees are able to implicitly represent this dimension, making use of the logarithmic nature of time in music, in the sense that note durations are multiples of basic time units, mainly in a binary (sometimes ternary) structure. This way, trees are less sensitive to the codes used to represent melodies, since there are less degrees of freedom for coding.

In this section the proposed tree construction method for representing a melody is presented, defining the terms needed to build the model. First, string representations are described as a reference. For all the discussions, the melodies are assumed to be monophonic: only one note can be played at a given time.

2.1 Pitch and duration representations

In string representations, note durations are coded with explicit symbols. For representing a melody as a string, symbols from a pitch description alphabet, Σ_p, and from that of duration, Σ_d are combined in $s \in \Sigma^*$, $s = \sigma_1\sigma_2...\sigma_{|s|}$. When these symbols are linked to those of pitch, the code is said to be coupled. In this case, $\Sigma = \Sigma_p \times \Sigma_d$, and σ_i will be a pair of pitch and duration descriptors. The pair for a note can only be formed when both dimensions are defined for it.

When both dimensions are handled independently, the representation is said to be decoupled or splitted. For decoupled string representations, $\Sigma = \Sigma_p \bigcup \Sigma_d$, being $\sigma_{2i-1} \in \Sigma_p$ and $\sigma_{2i} \in \Sigma_d$; $i = 1, 2, ..., \frac{|s|}{2}$. Similarly as before, the symbols for a note are included in the string only if both dimensions are defined for it.

Different kind of properties can be used for the symbols to represent the sequence of pitches in a melody [13, 14, 18]. They can be *absolute*, if the property depends only on the represented note, or *relative*, if it is defined in terms of differences to other notes, usually the preceding one. Next, some commonly used pitch properties

are presented. In each case, the alphabet, Σ_p, applicable is enunciated. The symbol '**s**' (for 'silence') denotes a rest.

Definition 2.1 *Pitch properties for each note:*

p1 **pitch name** *(absolute)* the name and octave. If notes are extracted from MIDI files, the alphabet is $\Sigma_{p1} = \{\mathrm{C}_{-2}, \mathrm{C}\sharp_{-2}, ..., \mathrm{F}\sharp_8, \mathrm{G}_8\} \bigcup \{\text{'}\mathbf{s}\text{'}\}$, $|\Sigma_{p1}| = 129$, although in practice is usually more reduced. The range for piano is $\{\mathrm{A}_{-1}, ..., \mathrm{C}_7\}$, which is enough for most cases. Using this range, $|\Sigma_{p1}| = 89$.

p2 **folded pitch** *(absolute)* the name without octave. $|\Sigma_{p2}| = 13$, corresponding to the 12 halftones of the octave, from A to G, including flat and sharp notes, and the rest.

p3 **pitch contour** *(relative)* $\Sigma_{p3} = \{-, =, +\}$; '+' if the pitch of the note is higher than that of the previous note, '−' if it is lower, and '=' if it is the same. As for the other relative pitch properties, for the first note in the sequence it is not defined. $|\Sigma_{p3}| = 3$.

p4 **high-definition contour** *(relative)* same as before but it also includes '+2' and '−2' if the pitch difference exceeds 4 halftones. $\Sigma_{p4} = \{-2, -1, 0, +1, +2\}$. $|\Sigma_{p4}| = 5$.

p5 **intervals** *(relative)* the difference in halftones between a note and the preceding one. Theoretically, $\Sigma_{p5} = \{i \in \mathbb{Z} \,|\, -127 \leq i \leq +127\}$, but in practice large intervals seldom appear, and some authors limit $\Sigma_{p5} = \{i \in \mathbb{Z} \mid -24 \leq i \leq +24\}$, being any other larger value assigned to the extremal values. This way, $|\Sigma_{p5}| = 49$.

Rests are not involved in the calculation of relative properties for the note following it, that are computed using the pitch of the note preceding the rest.

Similar definitions can be stated for the durations of the note sequence, defining a number of duration properties.

Definition 2.2 *Duration properties for each note:*

d1 **duration** *(absolute)* the difference between its onset and offset times, $t_{OFF} - t_{ON}$, usually expressed as multiples or fractions of the beat duration. Strictly speaking, this is not a numerable set, but in practice a limited set of durations appear.

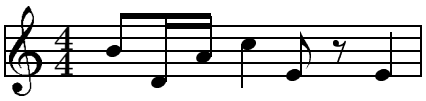
d2 **rhythm contour** *(relative)* $\Sigma_{d2} = \{-, =, +\}$; '+' if the duration of the note is longer than that of the previous note, '−' if it is shorter, and '=' if it is the same. For the first note in the sequence it is not defined. $|\Sigma_{d2}| = 3$.

d3 **inter-onset interval, IOI** *(absolute)* the time lapse from the ith note onset to that of the next; $IOI_i = t_{ON,i} - t_{ON,i+1}$, expressed as for *d1*. For the last note, *d3* is defined as its duration (*d1*). The same described about $|\Sigma_{d1}|$ is applicable for this case and the next. Note that rests disappear for this property.

d4 **inter-onset ratio, IOR** *(relative)* the ratio between successive IOIs; $IOR_i = \frac{IOI_i}{IOI_{i+1}}$. It is not defined for the last note and for rests.

Rest durations are treated the same way as notes for the properties *d1* and *d2*, but are ignored for the other two.

For the illustration of these properties, a simple melody has been displayed in figure 1 and coded in terms of these pitch and duration properties.



B3 D3 A3 C4 E3 s E3

PITCHES
p1: B3 , D3 , A3 , C4 , E3 , **s** , E3
p2: B , D , A , C , E , **s** , E
p3: - , − , + , + , − , - , =
p4: - , −2 , +2 , +1 , −2 , - , 0
p5: - , −9 , +7 , +3 , −8 , - , 0

DURATIONS
d1: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{4}$, 1, $\frac{1}{2}$, $\frac{1}{2}$, 1
d2: -, −, =, +, −, =, +
d3: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{4}$, 1, 1, -, 1
d4: 2, 1, $\frac{1}{4}$, 1, 1, -, -

Figure 1: Simple example of melody and how it is represented in terms of different pitch and duration descriptors. A short dash has been written when the code for a note is not defined.

Using the above defined descriptions, there are $5 \times 4 \times 2 = 40$ different ways of coding melodies as strings, being 5 the number of pitch codings, 4 the number of duration codings, and 2 corresponding to the coupled and decoupled way of combining both dimensions. Nevertheless, the ways of coding a melody as a tree using the proposed method are just 5, the number of different pitch descriptions defined above, since duration is implicit in the tree structure. In Fig. 2 some of these representations are displayed as an example. Note that the pair of symbols coding a note are only included when both are defined for it. For example, for the first note, interval (*p5*) and duration contour (*d2*) are not defined, or inter-onset properties (*d3* and *d4*) are not defined for rests.

p1 and *d1*, decoupled:	B_3 , $\frac{1}{2}$, D_3 , $\frac{1}{4}$, A_3 , $\frac{1}{4}$, C_4 , 1 , E_3 , $\frac{1}{2}$, **s** , $\frac{1}{2}$, E_3 , 1
p1 and *d1*, coupled:	(B_3,$\frac{1}{2}$) , (D_3,$\frac{1}{4}$) , (A_3,$\frac{1}{4}$) , (C_4,1) , (E_3,$\frac{1}{2}$) , (**s**,$\frac{1}{2}$) , (E_3,1)
p2 and *d2*, coupled:	- , (D,–) , (A,=) , (C,+) , (E,–) , (**s**,=) , (E,+)
p5 and *d3*, decoupled:	- , - , -9 , $\frac{1}{4}$, $+7$, $\frac{1}{4}$, $+3$, 1 , -8 , 1 , 0 , 1

Figure 2: Some string representations using different combinations of properties for the melody in Fig. 1. A short dash has been written when the code for a note is not defined.

2.2 Tree construction method

The tree representation approach proposed in this work is based on the fact that the duration of the music notation symbols are designed on a logarithmic scale: a *whole* note lasts twice a *half* note, whose length is the double of a *quarter* note, etc. (see Fig. 3).

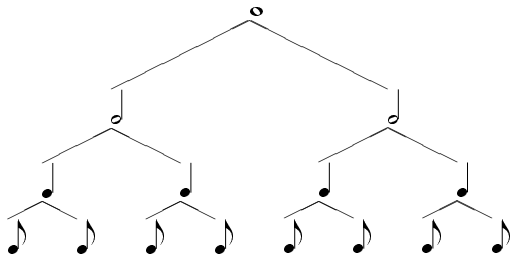


Figure 3: Duration hierarchy for different note figures. From top to bottom: whole (4 beats), half (2 beats), quarter (1 beat), and eighth (1/2 beat) notes.

Each melody measure is initially represented by a tree, τ_i. Each note or rest will be a leaf node. The left to right ordering of the leaves keeps the same time order of the notes in the melody. The level of each leaf in the tree determines the duration of the note it represents, as displayed in figure 3: the root (level 1) represents the duration of the whole measure (a *whole* note), each of the two nodes at level 2 represents the duration of a *half* note. In general, nodes at level i represent the duration of a $1/2^{i-1}$ of a measure.

During the tree construction, internal nodes are created when needed to reach the appropriate leaf level. Initially, only the leaf nodes will contain a label value, using the pitch properties described in definition 2.1, but then, a bottom-up propagation of these labels is performed to fully label the tree nodes. The rules for this propagation will be described later, in section 2.5.

An example of this scheme is presented in Fig. 4 using folded pitches as labels. In the resulting tree, the left child of the root has been splitted into two subtrees to reach the level 3, that corresponds to the first note (a quarter note, duration of a $1/2^2$ of the measure, pitch B). In order to represent the durations (both are 1/8 of the measure) of the rest and note G, a new subtree is needed for the right child in level 3, providing two new leaves for representing the rest (**s**) and the note (G). The half note (C) onsets at the third beat of the measure, and it is represented in level 2, according to its duration.

It can be seen in figure 4 how the order in time of the notes in the score is preserved when traversing the tree from left to right. Note how onset times and durations are implicitly represented in tree, compared to the explicit encoding of time needed by strings. Using the definitions 2.1, only five tree representations are possible, compared to the 40 for strings. In addition, this representation is invariant against changes in tempo, or different meter representations of the same melody (2/2, 4/4, or 8/8, for example).

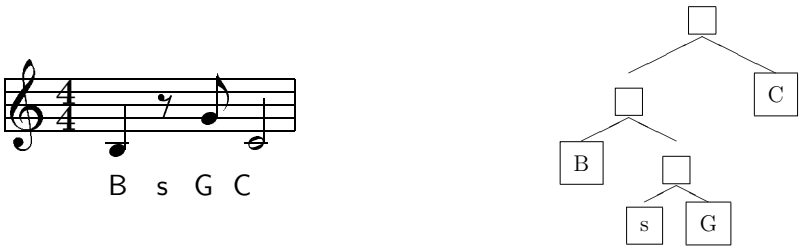


Figure 4: Simple example of tree construction with folded pitches (def. *p2*).

2.3 Processing non binary durations

In some occasions the situation can be more complicated. There are note durations that do not match a binary division of the whole measure. This happens, for example, for dotted notes (duration is extended in an additional 50%) or tied notes (their durations are summed) (see Fig. 5-left). In this situation, a note can not be represented just by one leaf in the proposed scheme. It is well known [19, 20] that our auditory system perceives in a similar way one note of a given duration and two notes of the same pitch, played one after the other, which durations sum that of the single one.

Thus, when a note exceeds the proper duration, in terms of binary divisions of time, it will be subdivided into notes of binary durations, and the resulting notes are

coded in their proper tree levels. In Fig. 5 an example of these situations is shown and how they are represented by this scheme.

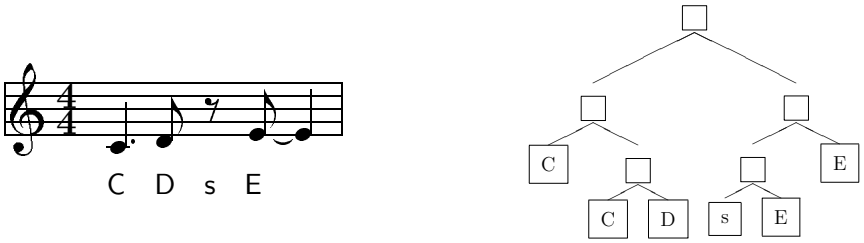


Figure 5: Tree representations of notes exceeding their notation duration: dotted and tied notes. Both 'C' leaves correspond to the same dotted quarter note. The two 'E' leaves represent the two tied notes.

Other frequently used non binary divisions are ternary rhythms. In that case, the length of one measure is usually 3 beats and it is splitted into 3 quarter notes, etc. This is not a problem, since neither the tree construction method nor the metrics used to compare trees need them to be binary, and the number of children for each node can be an arbitrary number. In ternary meters or ternary divisions, the number of children for a node will be three. This can be generalized to other more complicated cases that can appear in musical notations, like tuplets or compound meters. In figure 6 an example of compound meter based on ternary divisions and its representing tree is shown.

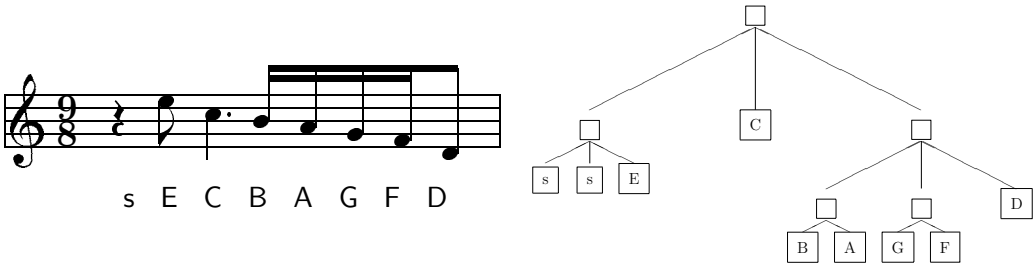


Figure 6: The meter 9/8 is a compound one based on ternary divisions. The tree construction method can represent this melody without problems.

There are other subtle situations that may appear in a score, like for example grace notes[1], that are not included in the cases described above. Anyway, in the

[1] A grace note is a very short note or a series or notes to achieve musical effects that occupies no time in the duration notation in a score. They also are named "acciaccatura".

digital scores, like MIDI files, these special notes do not appear, and short notes would be present for grace notes that will be coded in the level of the tree that corresponds to its actual duration. The details of these situations are described in detail elsewhere [11].

2.4 Representation of complete melodies

The method described above is able to represent a single measure as a tree, τ. A measure is the basic unit of rhythm in music, but a melody is composed of a series of M measures. Next, the way of combining the set of trees $\{\tau_i\}_{i=1}^{M}$ computed for every single measure is discussed.

Joining the set of computed measure trees in an ordered way is needed to build the tree, T, for the complete melody. For this purpose, a method for grouping the sub-trees is required. They can be grouped two by two, by adjacent pairs, repeating this operation bottom-up, hierarchically, with the new nodes until a single tree is obtained. Nevertheless, with this grouping method, the trees would grow in depth quickly:

$$h(T) = \log_2 M + 1 + \max_i \; h(\tau_i) \quad ,$$

making the tree edit distance computation very time consuming, as will be discussed in section 3.1. Another possibility is to build a tree with a root for the whole melody, being each measure sub-tree a child of that root. This can be considered as a forest of sub-trees, but linked to a common root node that represents the whole melody. This way, the tree depth for the whole melody will be only

$$h(T) = 1 + \max_i \; h(\tau_i) \quad .$$

This smaller depth of the whole melody tree, T, is a key point to choose this approach to build T. Figure 7-left displays an example of a simple melody, composed of three measures and how it is represented by a tree composed of three sub-trees, one per measure, rooted to the same parent node. The level 0 will be assigned to this common root.

2.5 Bottom-up propagation of labels and pruning

Two causes motivate the procedure described in this section. Firstly, tree edit distance algorithms need all the nodes (both internal and leaves) to have a label [17, 21]. After the structure of the tree is completed, the pitch labels are just in the leaves. A set of rules are used for propagating the labels from the leaves upwards, labelling the internal nodes. The propagation of a label is decided on the basis

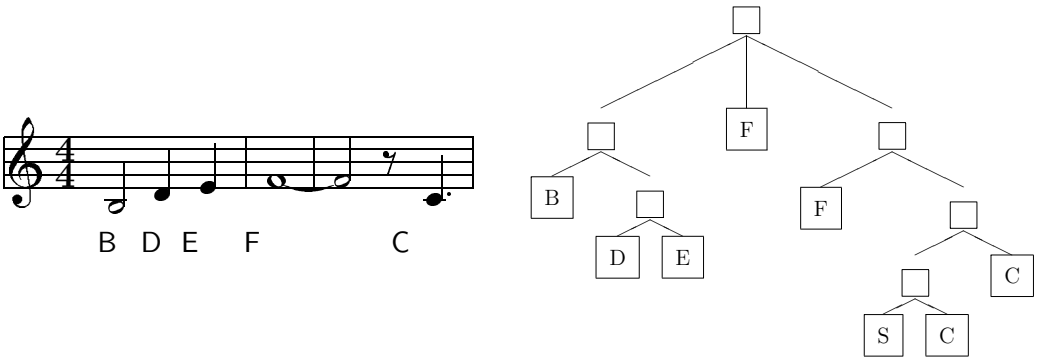


Figure 7: An example of the tree representation of a complete melody. The root of this tree links all the measure sub-trees.

that the note in that node is more important than that of the sibling node. The importance of a note is related to its capability to contribute to the melody identity.

Secondly, the high complexity of the tree edit distance computing (see section 3), requires the trees to be as small as possible. When very short notes appear or they do not match exactly the binary or ternary subdivisions of the beat, the resulting trees are very deep. Thus, the label propagation rules are accompanied of a pruning action to delete little significant branches when applying the rules below a given *pruning level*. This process also contributes to remove irrelevant information that would make the classification more difficult, obtaining reduced trees able to keep the main musical features of the melody.

Given a *pruning level*, p, the rules for propagating the labels to internal nodes and pruning the tree are defined below. In each case, a rule is applied to a sub-tree, and if the level l of the sub-tree is below the pruning level p, the labels are propagated and the tree is pruned; otherwise, the rule only propagates labels, keeping the structure of the tree. This pruning level is equivalent to the resolution desired for the resulting tree in terms of note lengths. This way, in the pruning tree, the notes represented will be always longer or equal to a $1/2^{p-1}$ fraction of the measure length. A value $p = \infty$ means that pruning is never applied.

The set of propagation (and pruning when applicable) rules are described below and illustrated in figure 8. For the definitions, a parenthesis notation is used for the trees, in such a way that a subtree, t, having a father node with label a, and two children: left with label b and right with label c, is denoted as $t = a(bc)$. If a node has not a label, we will consider it as labelled with the empty label, ϵ. All the labels, except ϵ, are symbols in one of the Σ_p alphabets. The value '**s**' is explicitly used for

rests.

All the definitions have been stated for binary sub-trees but they can be extended for ternary trees, keeping the meaning of each situation. The number of possible cases for each rule is much greater, so they have not been included here for clarity. All these rules are illustrated in figure 8.

Definition 2.3 *Propagation and training rules:*

r1 The *r1* rule simply propagates/prunes a unary tree:

$$r1\big[\epsilon(a)\big] = \begin{cases} a & \text{if } l \geq p \\ a(a) & \text{otherwise} \end{cases}$$

If there is only one child it is automatically upgraded. This situation seldom appears but it can be found in the rightmost note of an incomplete measure or building the tree from a single measure.

r2 The *r2* rule makes the pitch propagate over a rest:

$$r2\big[\epsilon(\mathbf{s}a)\big] = \begin{cases} a & \text{if } l \geq p \\ a(\mathbf{s}a) & \text{otherwise} \end{cases}$$

$$r2\big[\epsilon(a\mathbf{s})\big] = \begin{cases} a & \text{if } l \geq p \\ a(a\mathbf{s}) & \text{otherwise} \end{cases}$$

r3 The *r3* rule is also very simple, and joins equal pitches:

$$r3\big[\epsilon(aa)\big] = \begin{cases} a & \text{if } l \geq p \\ a(aa) & \text{otherwise} \end{cases}$$

If all the children of a node have the same label, they are deleted and its label is placed in the father node. Thus, two equal notes are equivalent to just one with double duration.

r4 If one of the children nodes has the same label as that of the father's sibling node, then the other label is propagated. This rule tries to avoid the propagation of a pitch that would be lost by the application of *r3* in the next step. This is formalized here for all possible cases:

$$r4\big[\epsilon(\epsilon(ba)b)\big] = \begin{cases} \epsilon(ab) & \text{if } l \geq p \\ \epsilon(a(ba)b) & \text{otherwise} \end{cases}$$

$$r4\big[\epsilon(\epsilon(ab)b)\big] = \begin{cases} \epsilon(ab) & \text{if } l \geq p \\ \epsilon(a(ab)b) & \text{otherwise} \end{cases}$$

$$r4\big[\epsilon(b\epsilon(ba))\big] = \begin{cases} \epsilon(ba) & \text{if } l \geq p \\ \epsilon(ba(ba)) & \text{otherwise} \end{cases}$$

$$r4\big[\epsilon(b\epsilon(ab))\big] = \begin{cases} \epsilon(ba) & \text{if } l \geq p \\ \epsilon(ba(ab)) & \text{otherwise} \end{cases}$$

r5 The *r5* rule limits the applicability of the former rules, that otherwise could propagate a very short pitch to a much longer note, eliminating other longer pitches. In order to avoid that, any rule (denoted as r in the rule below) can be applied only three times for the same label (this implies to stretch its length in a factor of 2^3 for binary meters).

$$r5\big[\epsilon(ab)\big] = \begin{cases} b & \text{if } l \geq p \\ b(ab) & \text{otherwise} \end{cases}$$

$$r5\big[\epsilon(ba)\big] = \begin{cases} b & \text{if } l \geq p \\ b(ba) & \text{otherwise} \end{cases}$$

iff

$$a \quad \text{is the root of} \quad t = r\big[r\big[r\big[\ldots\big]\big]\big]$$

and

$$a \quad \text{comes from a node 3 levels below.}$$

r6 The *r6* rule is a "default" case, whenever any other rule may be applied:

$$r6\big[\epsilon(ab)\big] = a(ab)$$

This rule upgrades the label of the left child, because in binary meters, the notes placed in odd beats are usually stressed. These notes are represented by left children in the tree.

All these rules are applied under certain conditions and precedence order that are described in the algorithm 1:

An example of the application of these rules is displayed in figure 9 with a level $p = 2$. One measure with some notes with different durations is considered. In the left side of that figure, the score and the tree as it results from the construction procedure is presented. The labels in that tree are folded pitches (*p2*) in definition 2.1).

In Fig. 9-left it can be observed how the propagation and pruning rules apply to the tree. A value of the pruning level $p = 2$ has been considered. This way, the rules applied below that level in the tree ($l \geq p$) are pruning rules, otherwise

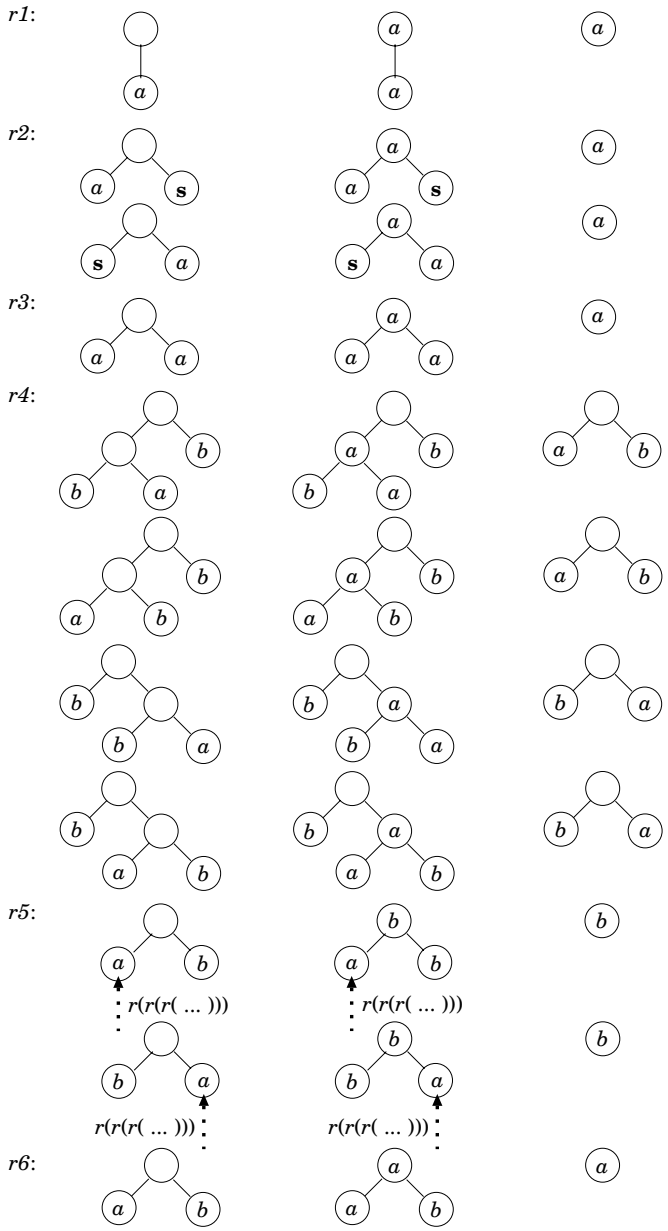


Figure 8: Propagation and pruning rules. (left column): original sub-tree; (center column) propagation rules; (right column): pruning rules.

Algorithm 1 Application of rules

if arity(t) = 1 **then**
  r1
else if left-child(t) = '**s**' or right-child(t) = '**s**' **then**
  r2
else if left-child(t) = right-child(t) **then**
  r3
else if root(t) comes from a leaf 3 levels below **then**
  r5
else if $t = \epsilon(\epsilon(ba)b)$ or $t = \epsilon(\epsilon(ab)b)$ or $t = \epsilon(b\epsilon(ba))$ or $t = \epsilon(b\epsilon(ab))$ **then**
  r4
else
  r6
end if

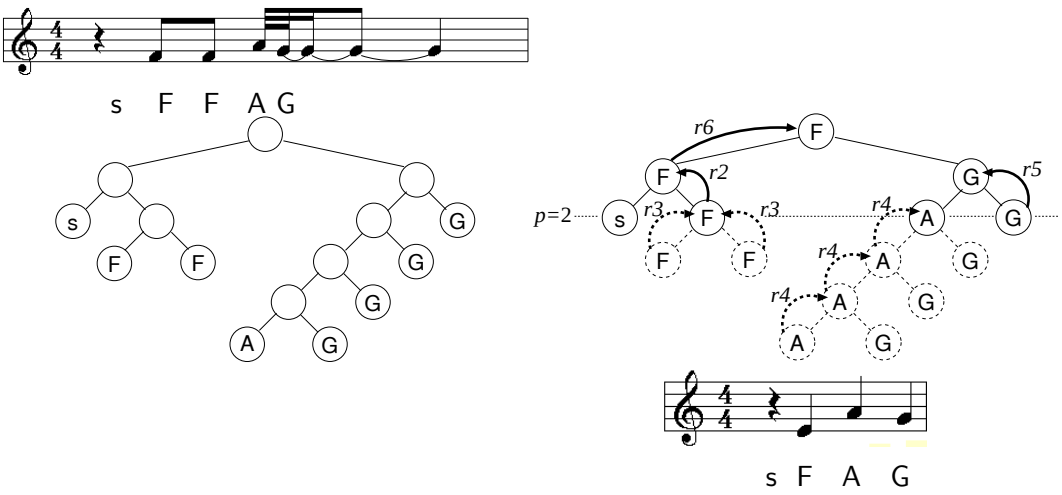


Figure 9: (left) One measure-melody and its tree representation with interval labels (only in the leaves now) before pruning and label propagation. (right) Final tree with propagated and pruned nodes (in dashed lines after applying dashed rules). The equivalent melody to the pruned tree is also displayed (right-bottom).

are just propagation rules. In the first half of the melody, the deepest levels have equal labels (F), so they are upgraded by the rule *r3* and then by *r2* because the sibling node is labelled with a rest. The second part shows how a very short note (A) is propagated by applying *r4* three times. Thus, *r5* is applied instead of *r6* that otherwise would have been applied, propagating 'A' again.

Note that in the score equivalent to that tree (Fig. 9-left-bottom) only quarter notes (in this context, their duration can be stated as a $1/2^{p-1}$ of the measure) have survived to the propagation and pruning rules, keeping the main features of the original melody.

3 Tree edit distance

Once the tree representation scheme has been introduced, the next section of this paper is for describing how the trees are compared. The problems that arise related to the complexity of this task are also discussed.

The edit distance between two trees can be defined as the minimum cost of the sequence of operations that transforms a tree into the other [17]. The editing operations are the same as those used in standard string edit distance (i.e. the Levenshtein distance): deletion of a node, insertion, and substitution of a node label. The more similar the structures of the trees are, the less operations of deletion and insertion have to be done, and the smaller the distance between them is.

3.1 Full edit distance

The Shasha and Zhang method [17] to compute the edit distance between two trees, T_A and T_B, has a time complexity of $O(|T_A| \times |T_B| \times h(T_A) \times h(T_B))$, where $|T_i|$ are the number of nodes in the trees and $h(T_i)$ are their depths. It uses the tree editing operations described above, giving a cost to each operation. The objective is to achieve a mapping between both trees that requires the least cost sequence of operations, looking for similar tree structures, that is, similar rhythmical patterns.

We have used the Shasha and Zhang algorithm [17] to compute the full tree edit distance. It obtains the distance between both trees that requires the least cost sequence of operations, looking for similar tree structures, that is, similar rhythmical structures.

The cost weights used for the edit distance operations have been set to 1 for insertion and deletion. For substitution, the weight is 0 if the label is the same and 1 otherwise. Other tested weights did not improve the results.

3.2 Partial edit distance

The high cost of the full edit distance, makes it advisable to look for a cheaper alternative. The technique introduced by Selkow [21] has this property. The main functional difference of this technique is that node insertions and deletions can be done only at the leaves of the trees. Only after removing all the subtree rooted at a node it can be deleted. The restriction of the way a node can be inserted or deleted makes the algorithm simpler, but less accurate.

The lower complexity is the main advantage of the Selkow method. Its time complexity is $O(n_A n_B h)$ where n_A and n_B are the maximum arities of the trees T_A and T_B, respectively, and h is the maximum depth of both trees. Due to the whole melody tree construction method described in section 2.4, joining all the measure sub-trees in a single, root, in our case, n_A and n_B will be the number of measures of the two melodies to be compared.

4 Nearest neighbour classification with tree edit distance

The NN classification rule is a widely known non-parametric technique for classification tasks. Although usually an object (*prototype*) is represented as a vector of features (a point in $\mathbb{R}^n$), the NN rule may also be used when objects are represented as strings or trees, if an adequate dissimilarity measure is defined.

When the distance has very high time complexity (like in our case), the classification time per sample becomes very high, if the exhaustive NN search is applied. As the bottleneck in this task is obviously the distance computation, a fast NN search algorithm is essential. More precisely, we need an algorithm that computes a very low number of distances, like AESA [22], LAESA [23], and TLAESA [24]. These algorithms are not the fastest when prototypes are represented as vectors, but do their best when distance computations are very time consuming, like when prototypes are represented as strings or trees, for instance, due to the small number of distance computations.

However, even with the algorithm that computes less distances (AESA), the average classification time per sample in our experiments was still too high, as we will explain later in section 5. In order to address this problem we have tried two alternatives: first, to use the Selkow tree edit distance, which is much faster but less accurate. Second, to extend previous work on approximate NN search [25], mainly focused on vector spaces of representation, to the algorithms mentioned above, which are suitable for any metric space, not only for vector spaces. We have also extended

the Fukunaga and Narendra's algorithm [26], which is also suitable for metric spaces in general.

4.1 Approximate NN search in non-vector spaces

The idea of approximate NN as stated in [25] is to find a neighbour of the unknown object (the *sample*) which is not farther than a certain factor ϵ from the actual NN of the sample, that is, its distance is not greater than $(1+\epsilon)d_{nn}$, where d_{nn} is the distance to the actual NN. The approximate NN search is thus faster than exact NN search when ϵ increases, but usually error rates also increase with the value of ϵ. Thus, the problem is to find an adequate value of ϵ in order to speed up classification without increasing too much the error rates.

In this work we present the application of the ideas in [25] to algorithms suitable for non-vector spaces; however, the changes needed for this task are algorithm-dependent. In the case of AESA and LAESA, the algorithms compute a lower bound of the distance of each prototype p to the sample x, $g(x,p)$, using the triangle inequality and some previously computed distances: given a set B of prototypes whose distance to the sample has been computed, and given that the distances from these protoypes to all other prototypes in the training set have been computed during the training of the classifier, the lower bound can be computed as:

$$g(x,p) \quad = \quad \max_{b \in B} |d(x,b) - d(b,p)|$$

In the case of AESA, the set B is the set of all prototypes whose distance to the sample has been computed (this implies a continuous reevaluation of the lower bound); however, a table holding all the distances between the prototypes in the training set has to be stored, thus the spatial complexity becomes quadratic. In the case of LAESA, the set B is selected at training time so that the prototypes in B are maximally separated, allowing an acceptable lower bound computation without the quadratic spatial complexity (see [22, 23] for the details).

Both algorithms traverse the training set, selecting a candidate to nearest neighbour as the one with the lowest lower bound. Then, its distance to the sample is computed and the current nearest neighbour is updated, if possible. The algorithm finishes the search when the next candidate c has a lower bound higher than the distance to the current nearest neighbour, d_{nn}, that is, when:

$$g(x,c) > d_{nn} \quad \text{(terminating condition for exact NN search)}$$

The extension of these algorithms for approximate NN search is straightforward:

$$(1+\epsilon)g(x,c) > d_{nn} \quad \text{(terminating condition for approximate NN search)}$$

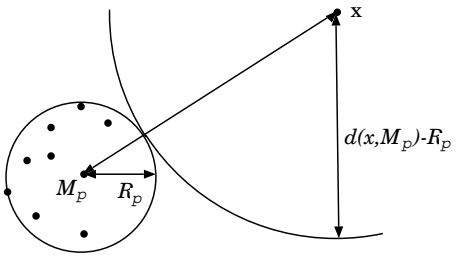


Figure 10: Lower bound of the distance from the sample x to a node p in the Fukunaga and Narendra's algorithm.

By using this new terminating condition the algorithm stops the search earlier (depending on the value of ϵ), thus allowing a faster classification.

The TLAESA and Fukunaga and Narendra's (FN75) algorithms both build up a tree from the training set and traverse it using a branch and bound scheme, similar to the tree traversal that uses the k-d tree, in which is based the approximate search proposed in [25]. One of the various implementations of approximate search over a tree uses a priority queue to store unvisited tree nodes. The nodes are stored along with a *key* m, which is used to sort the nodes in the queue, so that the node with the minimum key is the first to be extracted from the queue. In the case of TLAESA and FN75, the key for the priority queue is a lower bound of the distance from all the prototypes contained in a node p to the sample x (see figure 10):

$$\begin{aligned} m &= d(x, M_p) - R_p \qquad \text{(FN75)} \\ m &= g(x, M_p) - R_p \qquad \text{(TLAESA)} \end{aligned}$$

where M_p is the representative of the node, R_p is the radius of the node and $g(\cdot,\cdot)$ is a lower bound of $d(\cdot,\cdot)$ (computed exactly in the same way as in the LAESA algorithm). The expressions for the keys are derived from elimination condition for non-leaf nodes of each algorithm.

The search phase is very similar in both algorithms: at each step, the algorithm extracts a node from the queue (the one with the minimum key), and then it prunes one or both of its children and stores the others in the queue. Whenever the next node extracted from the queue has a key higher than d_{nn}, the algorithm finishes the search. When using a priority queue to traverse the tree, the approximate search is easy to incorporate: as in the case of AESA and LAESA, when $m(1+\epsilon) > d_{nn}$ the search finishes. In the four algorithms, letting $\epsilon = 0$ means an exact (non-approximate) NN search.

5 Experiments and results

The dataset has a total number of 641 prototypes (melodies extracted from MIDI files), from 149 different classes (different melodies). The MIDI files corresponded to film soundtrack themes, some well known pop-rock songs and classical music pieces from the "classical period", in such a way that different versions of those themes could be easily found in the Internet. Each melody prototype has been represented by a tree for each pitch property (producing 5 different trees) and by the 40 possible different string representations, as discussed in section 2.1.

The evaluation of classification error has been made using the leaving-one-out technique, due to the low number of prototypes per class available. All the classification experiments have been performed with the NN classifier.

The first experiment has been designed to assess the ability for melody identification of the different pitch properties described in definitions 2.1, and compare them to their use in strings together with the duration properties presented in definitions 2.1.

The second experiment tries to evaluate the classification and time performance of full and approximate tree edit distance, using approximate NN search. String representation performance will be taken as a reference also in these experiments.

5.1 Pitch representations

The performances for the five different kind of pitch properties as a function of the pruning level, $p \in [3, 8]$ and $p = \infty$ have been tested and compared to a number of string representations in terms of error rate and classification times.

The classification results for the pitch properties are plotted in figure 11. Intervals (*p5*) achieved the best performances, and a pruning level of $p = 5$ was the best in most cases. Note that strings performed worse than trees in general.

The error rates for all the considered representations were averaged for both trees and strings (dotted lines in figure 12). Average classification times (measured as time in seconds per prototype in the training set), computed with the full tree edit distance, are plotted in the same graph for comparison. Note that the string edit distance is much faster (around 0.3 s/prototype) than tree edit distance and this measure increases in time dramatically for $p \geq 5$. From the plots in that graph can be stated that $p = 5$ can be a good compromise between time and classification error, and this value will be utilised for the next experiments.

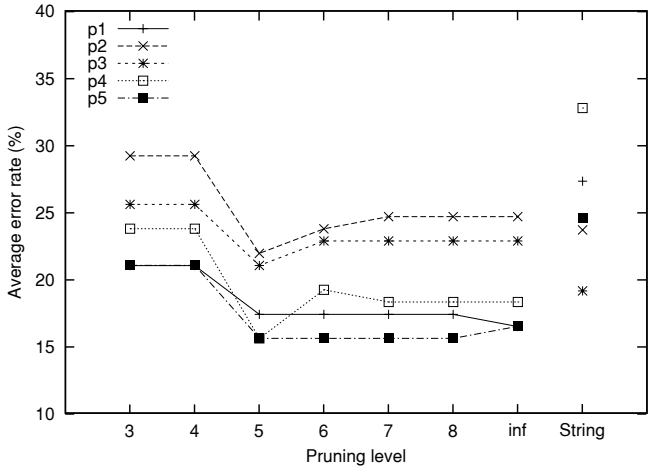


Figure 11: Classification performances with different pitch properties as a function of the pruning level. For clarity, the string results are displayed only for the duration property "duration" (*d1*) in a coupled coding with the different pitch properties.

5.2 Partial distance and approximate search

In this experiment, intervals have been used both for the tree representation and for the strings. For trees, the pruning level is $p = 5$. First, the performance of the full edit distance is studied versus the value of ϵ defined in section 4.1 and then, the same is done for the partial edit distance.

The error rates and average classification times per sample for the Shasha and Zhang distance are plotted in figure 13, for increasing values of ϵ, with the error rates and classification time for the string representation as a reference (without approximate NN search). The results show that using a tree representation improves the performance of the classifier, lowering its error rates around a 10 percent with regard to strings using the same pitch representation with the best of the four posible duration representations for that pitch. A value of $\epsilon = 2\%$ have been the maximum allowed in such a way that trees perform better than the best string coding.

The results depend highly on the NN search algorithm: the error rates and classification time for LAESA and TLAESA are very similar; in the case of AESA, its time performance is always the best, but has a quadratic space complexity that makes it useless for large training sets. The FN75 algorithm needs higher values for ϵ, as the other three algorithms compute the lower bound of the distance from a node to the sample. The best results are those of the AESA, but if we discard AESA due

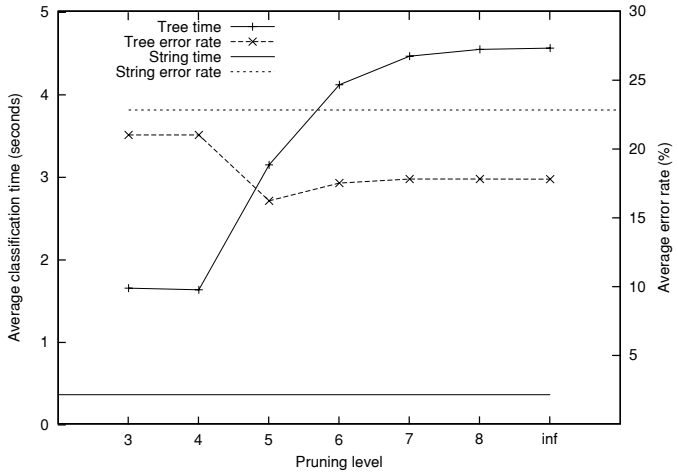


Figure 12: Evolution of time and error rate versus tree pruning level (averaged for all the different labels). References for strings are plotted as horizontal lines.

to its quadratic spatial complexity, all other three algorithms obtain similar results (although with different values of ϵ): if we allow approximately a 2 percent increase in error rates, the classification time may be reduced in more than a third part with respect to an exact NN search.

The classification times of Shasha and Zhang's distance are still too high with respect to the string representation, so we tested Selkow's tree edit distance. The same experiments were reproduced using this new distance, and the results are shown in figure 14. Although error rates increase a little with this distance, classification times have been dramatically reduced, and still it is possible to reduce them more using approximate NN search. Once again, the best results are obtained by AESA, but the results for the other algorithms are similar than those of the Shasha and Zhang distance. The important point is that the tree representation classification times are similar (and sometimes better) to those of the string representation, while the error rates keep lower than string ones.

6 Conclusions

Tree representation of melodies has been proposed to improve identification rates achieved by string representations and to reduce the degrees of freedom of strings for coding. To overcome the higher processing time of tree representation classifiers, a combination of low-cost partial tree edit distance and approximate NN search has

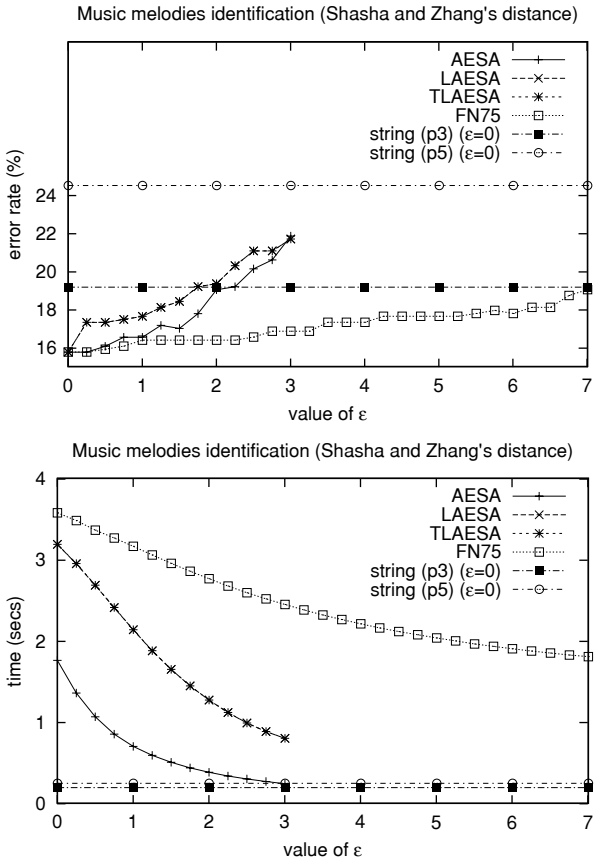


Figure 13: Error rates (top) and per-sample classification time (bottom) for the Shasha and Zhang distance. The error rates for a string representation are plotted as a reference.

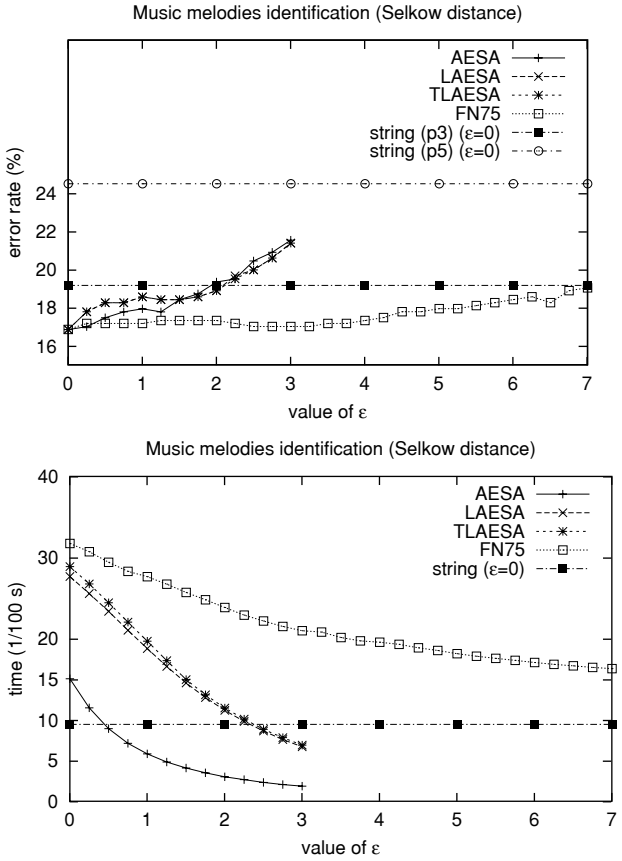


Figure 14: Error rates (top) and per-sample classification time (bottom) for the Selkow distance.

been proposed.

The results show that the tree representation reduces the error rates of string representations. The addition of rhythmic information to string coding in order to improve classification rates opens a high number of different possibilities that must be explored in order to reach the best possible result, while tree coding naturally represents that information in its hierarchical structure in a unique manner, thus reducing the degrees of freedom in the representation.

A set of rules have been defined in order to fully label the tree internal nodes and to prune the tree to keep its depth limited. Both things are needed to apply the tree edit distance and to reduce classification times. A maximum pruning depth equal to 5 (no notes shorter than an eighth note remain) provided small trees and good classification rates with our corpus.

Among all the pitch properties defined to label the trees, note intervals have produced the best results.

The combination of partial tree edit distance with approximate NN search allows the classification times to be comparable to or sometimes better than those of strings, with a small increase in error rates that still remain lower than those of strings.

For the future work we plan to develope and use methods for automatic motive extraction and segmentation of melodies. This would allow to extract melodic "thumbnails" that could be used as a representative of the whole melody for more efficient search and identification.

References

[1] Dongge Li, Ishwar K. Sethi, Nevenka Dimitrova, and Tom McGee. Classification of general audio data for content-based retrieval. *Pattern Recognition Letters*, 22:533–544, 2001.

[2] Brian Whitman, Gary Flake, and Steve Lawrence. Artist detection in music with minnowmatch. In *Proc. of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568. Falmouth, Massachusetts, September 10–12 2001.

[3] H. Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Seattle, Washington, May 1998.

[4] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. In *Proc. of the* 4^{th} *Int. Conf. on Music Information Retrieval, ISMIR*, pages 201–208, Baltimore, USA, 2003.

[5] Midi standard.

[6] MusicXML.

[7] P. P. Cruz and E. Vidal. Learning regular grammars to model musical style. In V.Honavar and G.Slutzki, editors, *Proc. of 4th. International Colloquium on Grammatical Inference (ICGI-98)*, pages 211–222. Springer-Verlag (LNAI Series), 1998.

[8] P. J. Ponce de León and J. M. Iñesta. Feature-driven recognition of music styles. In *Proc. of the 1st Iberian Conf. on Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*, volume 2652, pages 773–781, Majorca, Spain, 2003.

[9] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proc. of the Int. Conf. on Artificial Intelligence*, Las Vegas, USA, 2001.

[10] Efstathios Stamatatos and Gerhard Widmer. Music performer recognition using an ensemble of simple classifiers. In *Proc. of the* X^{th} *European Conf. on Artificial Intelligence ECAI*, pages 335–339, Lyon, France, 2002.

[11] David Rizo and José M. Iñesta. Tree-structured representation of melodies for comparison and retrieval. In *Proc. of the* 2^{nd} *Int. Conf. on Pattern Recognition in Information Systems, PRIS 2002*, pages 140–155, Alicante, Spain, 2002.

[12] David Rizo, José Manuel Iñesta, and Francisco Moreno-Seco. Tree-structured representation of musical information. In *Proc. of the* 1^{st} *Iberian Conf. on Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*, volume 2652, pages 838—846. Springer-Verlag, 2003.

[13] P. P. Cruz, E. Vidal, and J. C. Pérez-Cortes. Musical style identification using grammatical inference: The encoding problem. In Alberto Sanfeliu and José Ruiz-Shulcloper, editors, *Proc. of the 8th Iberoamerican Conf. on Pattern Recognition, CIARP*, pages 375–382, 2003.

[14] Shyamala Doraisamy and Stefan Rüger. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70, 2003.

[15] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Massachusetts, 1983.

[16] Carlos Agón, K. Haddad, and Gerard Assayag. Representation and rendering of rhythm structures. In *Proc. of the* 2^{nd} *Int. Conf. on Web Delivering of Music, Wedelmusic*, pages 109–116, Darmstadt, Germany, 2002. IEEE Computer Press.

[17] S. Shasha and K. Zhang. *Approximate Tree Pattern Matching. Pattern Matching Algorithms*, chapter 11, pages 341–371. Oxford Press, 1997.

[18] Y.E. Kim, W. Chai, R. Garcia, and B. Vercoe. Analysis of a contour-based representation for melody. In *Proc. of the Int. Symposium on Music Information Retrieval*, 2000.

[19] A.L. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In B. Smith and W. Eelsberg, editors, *Proc. of ACM International Multimedia Conference*, pages 235–240, Bristol, UK, 1998.

[20] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24:161–175, 1990.

[21] Stanley M. Selkow. The tree-to-tree editing problem. *Information Processing Letters*, 6(6):184–186, 1977.

[22] E. Vidal. New formulation and improvements of the nearest-neighbour approximating and eliminating search algorithm (AESA). *Pattern Recognition Letters*, 15:1–7, 1994.

[23] L. Micó, J. Oncina, and E. Vidal. A new version of the nearest neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing-time and memory requirements. *Pattern Recognition Letters*, 15:9–17, 1994.

[24] L. Micó, J. Oncina, and R. C. Carrasco. A fast branch and bound nearest neighbour classifier in metric spaces. *Pattern Recognition Letters*, 17:731–739, 1996.

[25] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998.

[26] K. Fukunaga and M. Narendra. A branch and bound algorithm for computing k–nearest neighbors. *IEEE Transactions on Computing*, 24:750–753, 1975.

Intestinal Motility Assessment with Video Capsule Endoscopy: Automatic Annotation of Intestinal Contractions

Fernando Vilariño, Panagiota Spyridonos, Jordi Vitri, Petia Radeva
Computer Vision Center. Universitat Autnoma de Barcelona.
Edifici 0. 08193 Bellaterra.
fernando@cvc.uab.es

Abstract

Intestinal motility assessment with video capsule endoscopy arises as a novel and challenging clinical fieldwork. This technique is based on the analysis of the patterns of intestinal contractions obtained by labelling all the motility events present in a video provided by a capsule with a micro-camera attached to it, which is ingested by the patient. However, the visual analysis of the video sequences presents several important drawbacks, mainly related both to the high amount of time needed for the visualization process, and the low prevalence of intestinal contractions in video. In this paper we propose a machine learning system to automatically detect the intestinal contractions in video capsule endoscopy, driving a useful but not feasible clinical routine into a feasible clinical procedure. Our approach is based on a sequential design with to basic aims: the reduction of the imbalance rate of the data set and the modular construction of the system, which adds the capability of including domain knowledge as new stages in the cascade. We provide a detailed analysis of the performance achieved by our system, showing a reasonable outcome in terms of several performance metrics.

Keywords: Video Capsule Endoscopy, Intestinal Motility, Classification, Imbalanced data sets

1 Introduction

Small intestine motility dysfunctions are shown to be related to certain gastrointestinal disorders which can be manifest in a varied symptomatology [1]. The analysis of the intestinal contractions of the small bowel, in terms of number, frequency and distribution along the intestinal tract, represents one of the methods with the highest clinical pathological significance [2], which has been successfully applied and reported in recent studies [3]. Certain myopathic diseases have been associated

with functional abnormalities of the intestinal muscle, which carry the presence of weak intestinal contractions and gastrointestinal dysfunctions. Other pathologies have been shown to be related to neuropathies which affect the way the nervous system controls the intestinal activity, presenting intestinal motility disorders that lead to disorganized contractions and hinder the movement of the nutrients along the intestinal tract. Ileus, bacterial overgrowth and the irritable bowel syndrome have been reported as major clinical disorders related to intestinal dysmotility; in these cases, the analysis of intestinal contractions has shown to be a helpful tool, as well [2].

Current techniques for assessment of small intestinal motility are multiple and complementary [2, 4], but small intestinal manometry is widely accepted as the most reliable so far. Small intestinal manometry technique consists of the measurement of the pressure in certain points of the small intestine by means of multiple pressure sensors distributed along a thin tube that is introduced through the esophagus, giving as a result a graph with the contractile activity presented as variations in pressure detected by the sensors. Two main sets of drawbacks are associated with this technique: On the one hand, it is an invasive test which carries discomfort problems for the patient, and the presence of medical staff is needed throughout the whole process. On the other hand, its clinical value is limited to the examination of severe intestinal motor alterations, and it suffers a lack of sensitivity over certain types of intestinal contractions that cannot be detected by means of this method.

In this paper, we address the study of intestinal contractions in a novel approach using Wireless Capsule Video Endoscopy (WCVE) as data source. WCVE consists of a capsule with a camera, a battery and a set of led lamps for illumination attached to it, which is swallowed by the patient, emitting a radio frequency signal that is received in an external device. The result is a video movie which records the trip of the capsule along the intestinal tract with a rate of two frames per second, and that can be easily downloaded into a PC with the camera software installed. This technique overcomes most of the drawbacks related to manometry: it is much less invasive, since the patient simply has to swallow the pill, which will be secreted in the normal cycle through the defections; moreover, there is no need of hospitalization nor expert support through the process and the patient can lead an ordinary life, since the attached device is recording the video movie emitted by the pill. Once the video is downloaded into the workstation, the expert visualizes the zone of interest and labels those frames where an intestinal event is detected, obtaining the temporal pattern of intestinal contractions which is to be used as a base for the intestinal motility dysfunction assessment. However, the visualization and precise interpretation of the capsule recordings is not straightforward, but it is time consuming and stressful, since the prevalence of contractions in video is very low (1:50 frames). Visualization

time can vary depending on the frame ratio used for this purpose, but generally speaking it is common that for a visualization study of the whole small intestine the expert takes about one or two hours, making it not feasible as a clinical routine.

In order to deal with the drawbacks we have mentioned above, and make the analysis of the information provided by the capsule feasible for clinical routines, we have focused our efforts on the design of a system for the automatic annotation of intestinal contractions in capsule video endoscopy. Several works have been reported in the fieldwork of classical endoscopy, addressing the support of automatic systems for the diagnosis of different pathologies, such as ulcer or cancer, with applications based on digital image analysis and processing. In these studies, endoscopic images have been analyzed in terms of textural, color and other morphological features [5–10]. As far as we know, no preceding work has been reported on computerized analysis of capsule video endoscopy for the automatic identification of specific motility events such as intestinal contractions, making this novel framework a challenging and open field of research.

Our proposal is based on a machine learning system which automatically learns and classifies contractions from a capsule video source, providing the expert with a subset of the video sequences which are highly likely to contain intestinal contractions. This yields to a considerable reduction in visualization time, and the consequent reduction of stress, since most of the sequences to be analyzed are real contractions. In addition, one of the main advantages of our system is related to its ability to dynamically adapt itself to the different patterns of intestinal activity associated with intestinal contractions.

The rest of the paper is organized as follows: In Section 2, we develop the analysis and explanation of WCVE images, the different visual appearance of the different types of intestinal contractions and the difficulties inherent in their detection. In Section 3, we describe the structure of our video analysis system, namely, the feature extraction and the classification stages. Section 4 presents our experimental results. Finally, we devote the last sections of this paper to the analysis and discussion of the results provided, and the exposition of our proposals for future pieces of research on intestinal motility with video capsule endoscopy.

2 INTESTINAL CONTRACTIONS IN VIDEO CAPSULE ENDOSCOPY

2.1 Basic concepts on gastrointestinal motility

Muscle layers of the gut wall and their innervation are organized to provide the motor functions along the intestinal tract. The interaction of the gut with the central nervous system is performed through either somatic or autonomic neurons, and communication between various parts of the gut is performed by the transmission of myogenic and neurogenic signals longitudinally along it, as well as reflex arcs transmitted through autonomic neurons [1]. As a result of muscular stimulation, a contractile activity and tone is produced, and intestinal contractions are generated. From a physiological point of view, the different patterns of contractions can be gathered into two main categories, namely, *phasic* and *tonic*. The former are characterized by a sudden closing of the intestinal lumen, followed by a posterior opening, while the latter corresponds to a sequence of a closed lumen for an undetermined span in time. Both the type and the spatial frequency of intestinal contractions depend on the region of the gastrointestinal tract (stomach, small intestine or colon), and the temporal patterns they present are different during fasting (before the ingestion of nutrients) and postprandial stage (after the ingestion of nutrients). In this work, we restricted our field of research to the study of small intestinal motility assessment by means of the analysis of phasic contractions in fasting patients, in an attempt to provide a first approach to the global problem. The further extension of this work to tonic contractions, and the generalization of intestinal motility assessment by means of the analysis of postprandial patients is part of our current line of work, and it constitutes the object of study for subsequent pieces of research.

2.2 Intestinal contractions sequences with capsule endoscopy

Video capsule endoscopy images show a perspective of the inner gut during the trip of the capsule along the intestinal tract. This modality of images present a circular field of view, in which the intestinal wall and the intestinal lumen are shown (see Figure 1). The contraction of the muscle layer of the intestine is observed as a closing movement of the lumen, which is spanned over a few frames in the case of phasic contractions, and a longer sequence for tonic. Figure 2 shows a mosaic where the frames of a video have been deployed in a sequential way and different intestinal contractions have been outlined in a green frame. In order to completely understand the visual paradigm of phasic contractions in video capsule endoscopy, important physiological and technical issues have had to be taken into account: On

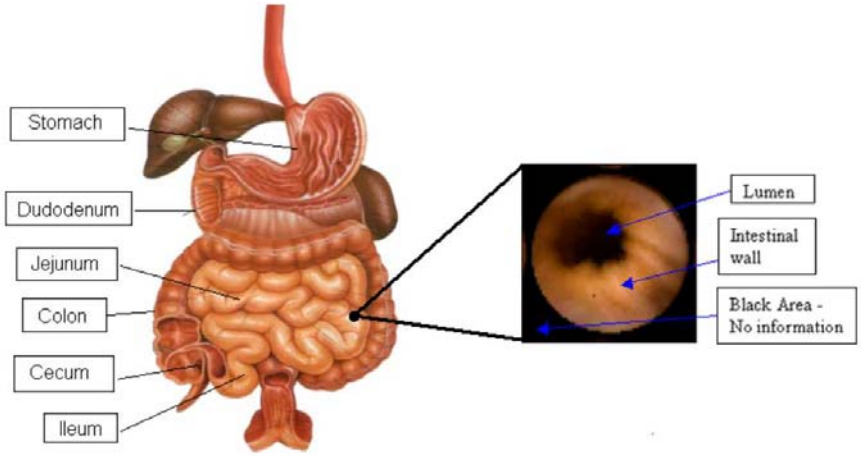


Figure 1: Appearance of a frame in capsule video endoscopy. The intestinal lumen and walls are rendered in a circular field of view.

the one hand, the maximal frequency of phasic contractions is known to be between 11 and 12 events per minute, spanning 4 to 5 seconds in average for the whole open-close-open cycle [1, 2], while the frame acquisition ratio of the camera is typically set on 2 frames per second [11]. Thus, we adopted the convention of bounding the span of a phasic contraction for fasting videos in a sequence of 9 frames. In the rest of the paper, we refer to a contraction sequence as a 9 frames sequence, where the central frame is set to be the frame that will be labelled as a detected contraction (tonic contractions might be treated in a different way, both for the differences in their duration and their separate physiological implications). On the other hand, the intensity with which the intestinal walls concentrically contract is not the same for all the contractions, and sometimes the closing of the lumen is not complete. If the lumen is completely closed during a contractile activity, this kind of event is referred to as an occlusive contraction; in case the lumen closing is not complete, the intestinal contraction is referred to as non occlusive. Non occlusive contractions are hard to detect by classical manometry, since the intestinal walls are not accomplishing enough amount of pressure to the pressure detectors. In video capsule endoscopy this kind of contractions is clearly shown, however. Figure 3 pictures out two sets of three different examples of (a) occlusive and (b) non occlusive contractions.

Unfortunately, the visual patterns of intestinal contractions in capsule endoscopy are not usually as clear as those rendered in Figure 3. The origin of this variability is twofold and strictly related to 1) technical issues linked to the movement of the

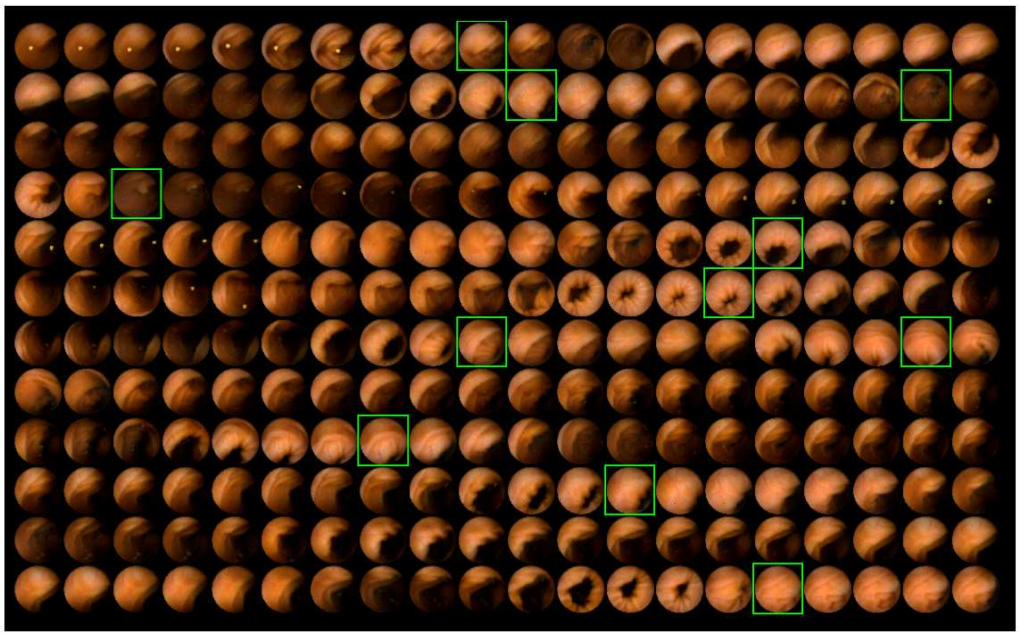
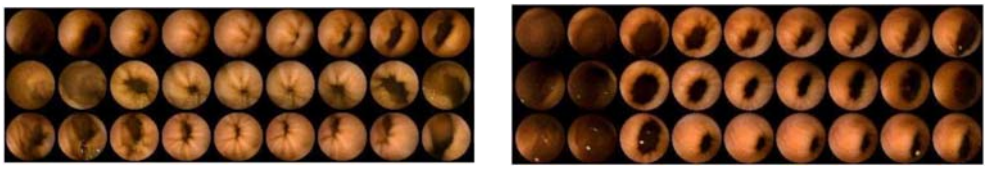


Figure 2: An example of capsule endoscopy video, which has been sequentially deployed for visualization purposes. Some intestinal contractions can be distinguished surrounded by a green square.

capsule device along the bowel, and 2) other physiological reasons -see Figure 4 for a graphical representation-:

1. *Camera movement*: The position of the camera in the intestinal lumen during the contractile activity is not steady. Since the capsule is freely moving into the gut, multiple changes in direction (namely, focusing the intestinal lumen or the lateral intestinal wall) and orientation (i.e., facing the proximal or distal parts of the tract) are performed. As a result, the camera is not always focusing the central part of the lumen, and this yields to a high variability of the visual patterns obtained in the video sequences. Figure 5 renders a representative set of examples of this situation in three intestinal contractions labelled by the specialists.

2. *Turbid liquid*: The good visibility of the intestinal lumen and wall is usually hindered by the presence of intestinal juices mixed up with the remains of food. This is visualized as a semi-opaque turbid liquid in a wide range of colors from brown to yellow. In addition to this, the turbid liquid may be



(a) Occlusive (b) Non-occlusive

Figure 3: Three sequences of nine frames of occlusive and non-occlusive intestinal contractions. The central frame corresponds with the frame labelled by the expert identifying the contraction event.

accompanied by the presence of bubbles and other artifacts related to the flux of the different liquids into the gut. As a result, the turbid liquid is interposed between the camera and the intestinal contraction event, obstructing its right visualization. This phenomenon is shown to be more acute in the case of postprandial studies, but it is relevant for fasting videos, as well. Figure 6 shows some example sequences of contractions which include the presence of turbid liquid.

In order to tackle the inherent complexity associated to the diverse visual patterns which an intestinal contraction may manifest in video capsule endoscopy, we focused our efforts on the design of a machine learning system for the automatic labelling of intestinal contractions. Through the next section we describe and justify the main traits of our approach, which is constructed in a sequential way following the shape of a cascade.

3 A cascade system for the detection of intestinal contractions in video capsule endoscopy

Our system is deployed in a sequentially modular way, namely, a cascade, as pictured out in Figure 7. Each part of the cascade receives as an input the output of the previous stage. The main input consists of the video frames, and the main output consists of the frames suggested as contractions. The rejected frames are distributed among three different stages: a first threshold stage, where most of the non-contractions frames are filtered; a second stage, where turbid not valid for analysis frames, wall frames and tunnel frames are rejected; and a final classification stage based on a support vector machine, where the final output is provided as sug-

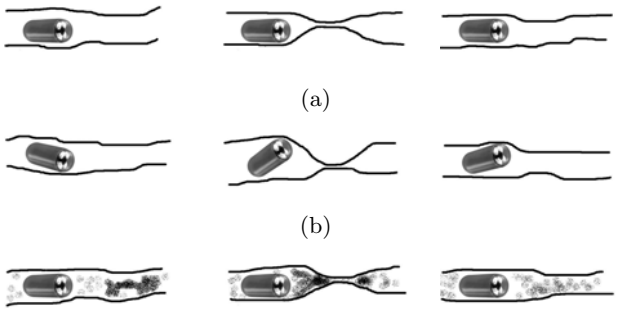


Figure 4: Graphical representation in three steps (before, in the moment of, and afterwards the contraction event): (a) The paradigm of a phasic contraction, (b) the camera pointing towards the intestinal wall, and (c) the presence of turbid liquid hindering the visualization. These patterns match the sequences rendered in Figures 3, 5 and 6 respectively.

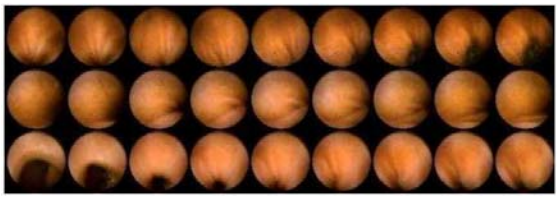


Figure 5: Three sequences of intestinal contractions showing diverse patterns due to the random camera orientation regarding the intestinal lumen.

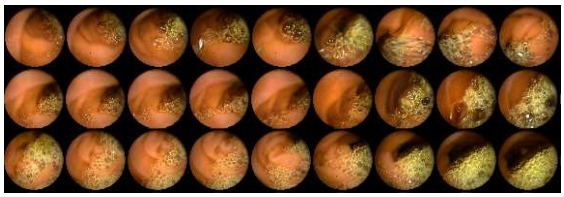


Figure 6: Three sequences of intestinal contractions with presence of turbid liquid, which hinders the correct visualization of the event.

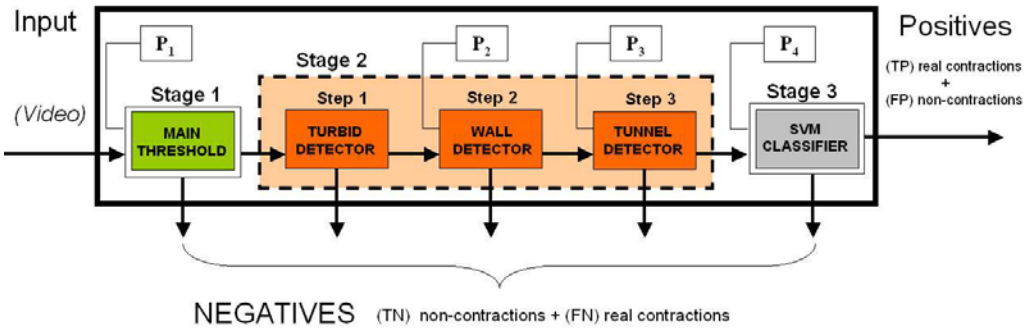


Figure 7: Cascade system for intestinal motility assessment. The input is the video studio and the output are the intestinal contraction frames suggested by the system. Each stage, rejects sequences of non-contractions. The global performance can be tuned by the set of parameters P.

gested contractions. The learning steps of each stage of the cascade involve a set of parameters P for tuning the classification performance. The turbid frames step and the final classification step consist of two support vector machine classifiers trained with a data set which has been labelled from previous studies.

The choice of the cascade system is underpinned by the fact that each step is designed in order to reject an amount of frames which mainly include images which are not to be intestinal contractions -i.e., the system negatives-, letting pass through the sequential stages those frames related to intestinal contractions -i.e., the system positives-. This yields to an effective reduction of the imbalance ratio of the data set at the input of the last classification stage. Many authors have applied diverse strategies in order to tackle the impact the imbalance ratio has in the performance of classification, involving stratified sampling, cost-sensitive approaches, different implementations of decision trees and bagging, and the use of several metrics for performance measurement, mainly [12–17]. In our strategy, each stage is tuned to prune as many non-contraction frames as possible, trying to minimize the loss of true positives, and achieving in this way an effective reduction in the imbalance ratio of the data. The last stage of the cascade, consisting of the support vector machine classifier trained by means of under-sampling, is to face a classification problem with an imbalance ratio about 1:5 -in contrast with the 1:50 at the input of the system-. This reduction in the imbalance ratio is shown to be an effective way of tackling the problem of classification in this kind of scenarios. In addition to this, one more important feature must be outlined: the modular shape of the system lets the expert identify new targets in the video analysis procedure, providing the chance to easily

include them as new filter stages, and adding domain knowledge to the system in a natural and flexible way. Through the following paragraphs, we provide a detailed description of each specific stage of the cascade.

3.1 Stage 1: Main threshold

The aim of the first stage is to pre-filter all the video frames according to the visual pattern of phasic contractions described in section 2.2. This is implemented by means of the normalized intensity $f^1(n)$, defined in equation 1.

$$f^1(n) = I_n - \frac{\sum_{i=-4}^{4} I_{n+i}}{9} \tag{1}$$

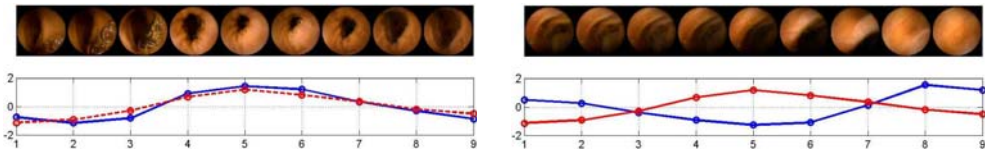
For each frame n, we take into account the 4 previous and the 4 following frames. For each one of these 9 frames, we calculate the overall intensity, $I_{n+i}, i = -4...4$, as the sum of the intensity values of its pixels. The final value of $f^1(n)$ represents a normalized intensity of the central frame within the 9 frames sequence. Should the central frame n be darker than its neighbors, the difference in $f^1(n)$ would tend to be negative, and viceversa. For the specific visual pattern of phasic contractions, the presence of an open lumen in the previous and following frames makes the central frame of a sequence of an intestinal contraction have a higher value of intensity than its neighbors. Thus, $f^1(n)$ is designed in order to present a high value when the central frame of a 9 frames sequence corresponds with an intestinal contraction, presenting a random pattern for non-contractions. A plot of $f^1(n)$ for (a) one contraction sequence and b) one arbitrary sequence of 9 frames is pictured in Figure 8. This first stage has got one tuning parameter P_1 associated to it, namely, the threshold value from which a sequence is to pass to the next stage.

3.2 Stage 2: Rejection of turbid, wall and tunnel frames

The aim of stage two is to reject those frames which are to be not valid for analysis according to the specifications described in section 2.2, namely, the turbid frames and those frames where the camera is focusing on the intestinal wall. In addition to this, those frames where the lumen appears static for a long sequence of time are rejected as well, as these frames do not carry out motility information.

3.2.1 Turbid frames

Turbid frames are those where the presence of turbid liquid hinders the right visualization of the lumen, and consequently, no motility information can be inferred



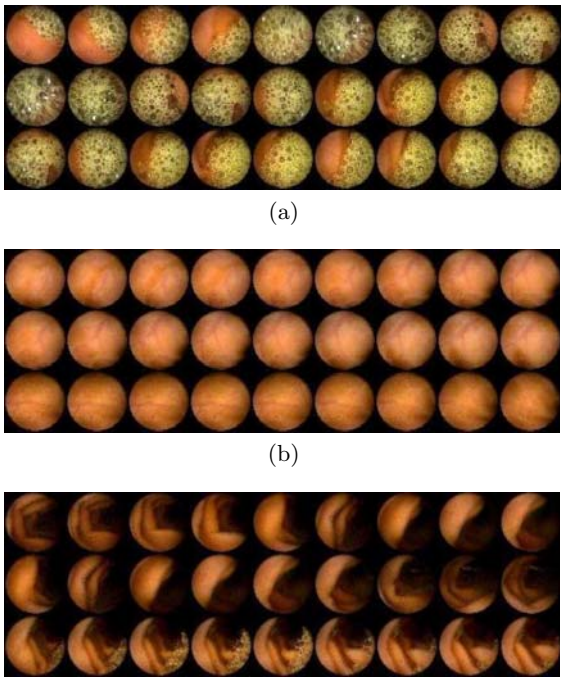
(a) Contraction (b) Random sequence

Figure 8: Pattern of f^1 for (a) one contraction and (b) one random sequence (solid blue line). The dashed red line corresponds to the averaged pattern of all the labelled intestinal contractions. The stage one filters sequences of frames applying a threshold over f^1.

from them. The presence of turbid liquid is characterized in terms of color, which is usually in a range from brown to yellow, mainly centered around green. For each frame, a color quantization is performed in the following way: each RGB component of the image is quantized into 5 bins in a linear way, spanning all the range of the color component. This yields to a 125-bins histogram (5^3), which is used as a feature vector. A data set of characteristic turbid frames and an under-sampled number of non-turbid frames are randomly chosen from a pool of reference studios, and they are used to train a support vector machine classifier (SVM) [18]. The SVM precise two main generalized parameters to be set: the kernel type and the kernel parameter. We used a radial basis function kernel and a $\gamma = 0.1$. Equation 2 shows the mathematical representation of the radial basis function kernel.

$$K_{rbf}(x, x_i) = \exp \frac{-|x - x_i|^2}{2\sigma^2}, \gamma = 1/(2\sigma^2) \tag{2}$$

The choice of the kernel and the γ parameter was obtained in a heuristical way with an exhaustive analysis, using as a reference for validation the visual assessment of the experts. The support vector machine classifies all the video frames into turbid and non-turbid. In order to incorporate the dynamic characteristics of the intestinal contractions as performed in the first stage, we adopted as a final criterion the rejection of those frames with more than 4 neighbors labelled as turbid frames within the 9 frames sequence (the number of 4 frames was strictly based on the experts' assessment), letting the remaining frames pass to the next step. Figure 9(a) renders a set of example sequences rejected as turbid sequences.



(a) (b) (c)

Figure 9: Three example sequences of (a) turbid, (b) wall, and (c) tunnel frames. The system detects and rejects these sequences as system negatives in the second stage.

3.2.2 Wall and tunnel frames

Both wall and tunnel frames are characterized by not carrying out motility information, although they have different sources of origin. The former are due to the stable orientation of the camera towards the intestinal wall, keeping the intestinal lumen out of the field of view, while the latter correspond to a stable orientation of the camera focusing the intestinal lumen for a span of time where no motility activity is present (in this sense, the resulting sequences show the intestinal lumen as a tunnel, during a undefined period of time). Figure 9 shows three different examples of (b) wall and (c) tunnel sequences. Both wall and tunnel frames were described by means of the sum of the area of the lumen throughout the sequence of 9 frames. In order to estimate the area of the lumen in each frame, a Laplacian of Gaussian filter was applied (LoG) [19]. The LoG filter is a second order symmetric filter with a tuning parameter σ which plays the role of a scale parameter. The output of

the LoG is high when a dark spot is found, providing a higher response the closer the diameter of the spot is fitting the span of the Gaussian defined by σ, and the higher the contrast is between the dark spot and its bounds. The value of σ was fixed to $\sigma = 3$, the minimum size of the lumen in the central frame of a contraction sequence (this was straightforward to obtain after testing different values of several contraction sequences). The whole procedure is graphically deployed in Figure 10: For each sequence of 9 frames, the LoG filter is applied (second row); following, a greater-than-zero threshold is performed to the filter output, which provides a binary image with one or more connected components or blobs (third row). In case that only one blob is obtained, its area is taken as the lumen area; in case that several blobs are obtained, the one with the highest global response of the filter (i.e., presumed the one with the highest contrast and best fitting in size) is selected. The last row in Figure 10 shows an example of the lumen segmentations obtained with this procedure.

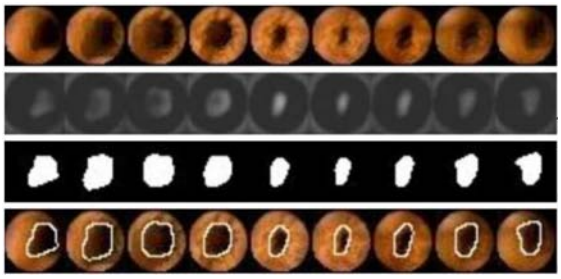


Figure 10: Original image, LoG filter response, binary blob and final lumen segmentation for the nine frames of an intestinal contraction sequence.

The subsequent characterization of wall and tunnel frames is straightforward: the system classifies a frame as a wall frame if the sum of the lumen area throughout the 9 frames sequence is lesser than a certain threshold, while the same frame is classified as a tunnel frame if the sum of the lumen area throughout the 9 frames sequence is greater than certain threshold. These two values yield to the tuning parameters of the system P_2 and P_3.

3.3 Stage 3: The final classifier

The last stage of our approach consists of a SVM classifier, which receives as an input the output of the second stage of the cascade, with an imbalance ratio which has been typically reduced form $1:50$ to $1:5$ frames. The output of the support vector classifier consists of frames suggested to the specialist as the candidates for intestinal

contractions in the analyzed video. The choice of the SVM is underpinned by its robust mathematical background, being one of the most widely used classification techniques, with a remarkable success in multiple and diverse applications through the recent years [20]. In addition to this, one of the main considerations taken into account for the selection of the SVM classifier was its sensitivity to the skewed distribution of the data sets. It has been shown that the learning mechanisms of SVM makes this classifier an attractive candidate for dealing with moderated imbalanced ratios. The SVM takes into account samples which are close to the decision boundaries, namely, the support vectors, and it tends to be unaffected by samples lying far away. In addition to the former, stratified sampling techniques (such as under-sampling the majority class, over-sampling the minority class, or artificially creating new samples) have been proved to be efficient in the improvement of performance of several classifiers, including support vector classifiers [21]. Our approach implements under-sampling in the learning strategy. Several methods of sampling were tested, and under-sampling resulted the one with the highest reliability (a detailed analysis and discussion about the design of these experiments can be found in [22]). A radial basis function kernel was used with a $\gamma = 0.01$ set in a heuristical way. The γ parameter controls the operation point of the support vector classifier, and corresponds to the fourth tuning parameter of the system, P_4.

In order to characterize the intestinal contractions, a set of 37 features were computed from first and second order statistics obtained from the concurrence matrix [23], and local binary patterns [24]. These features were estimated from a more basic feature set such as the mean intensity described in section 3.1, and the area and contrast of the lumen, as described in section 3.2.2. As performed in the previous stages, a feature vector was constructed taking into account the previous and following 4 frames, so that a final $37x9 = 333$ dimensional feature vector was assigned to each frame. In order to address the high dimensionality of the feature space, a sequential forward feature selection method was used based on the performance of the SVM.

4 Results

Our experimental tests were performed using 10 videos obtained from 10 different fasting volunteers (without eating or drinking in the previous 12 hours to the studio), aged between 22 and 33, at the Digestive Diseases Dept. of the General Hospital de la Vall D'Hebron in Barcelona, Spain. The endoscopic capsules used were developed by Given Imaging, Ltd., Israel [25]. The capsules dimensions were 11x26 mm, contained 6 light emitting diodes, a lens, a colour camera chip, two batteries with a mean life of about 6 hours, a radio frequency transmitter, and an antenna. The capsule

acquisition rate was two frames per second with a resolution of 256x256x24-bit. For each studio, one expert visualized the whole video and labelled all the frames showing intestinal contractions between the first post-duodenal and the first cecum images. These findings were used as the gold standard for testing our system. The parameter vector P was set to $P^0 = \{P_1 = 0, P_2 = 50, P_3 = 650, P_4 = 0.01\}$ using an exhaustive heuristical search, as defined in Section 2.2. Performance results were evaluated for each studio following the leave-one-out strategy: one video was separated for testing while the 9 remaining videos were used for training the SVM classifiers using under-sampling.

4.1 System performance

In order to accomplish a detailed system performance analysis of our approach, we provide the study of each separate stage in the cascade. Tables 1, 2 and 3 show the performance results of stages 1, 2 and 3, respectively. The meaning of the columns shown in these tables deserves a preliminary explanation: Each stage is viewed as a black box with both an input and an output. For each stage, certain number of frames arrive at the input (column **Frames**), containing a number of intestinal contractions labelled by the expert (column **Findings**); this yields to certain imbalance rate at the input of the stage, calculated as the quotient of the non-contraction frames over the number of findings (column **Imb. Rate**). The output columns consist of the number and the percentage of frames and findings passing to the next stage, and the resulting imbalance rate. In addition to this, the rate of lost findings, i.e., findings which were wrongly filtered as non-contractions, and the rate of non-contractions frames, i.e., non-contractions which were wrongly detected as contractions, is provided. The following paragraphs deploy a detailed analysis of each stage, paying special attention to the reduction in the imbalance rate and the accuracy of the classification performed.

- **Stage 1:** As we stated above, the primary aim of stage one was to pre-filter as many frames as possible, reducing the imbalance rate without a significant loss in contractions. Table 1 shows that the overall number of frames at the output of stage one is about 11% the input, i.e., the system rejects 89% of the frames in this stage. But despite this high reduction in the number of frames, almost every finding was kept (97%), i.e., just about a 3% of the findings were wrongly rejected as non-contractions. At the output of stage one, the imbalance ratio was reduced about 10 times, from 61.3 to 6.9.

- **Stage 2:** A similar analysis can be performed for stage two. At its output, this stage rejected about 28% of the frames, keeping the 96% of the findings

provided by stage one. The imbalance rate was reduced to 4.6. In addition to this, the sum of the loss of findings, taking into account both stage one and stage two, set the rate of detected contractions at the output of stage two about 93%, as can be observed in the column **%Findings in video** in Table 2. As in the previous stage, the reduction of the imbalance rate is significant, while the loss in contractions appears to be reasonable -only 7% of all existing contractions in video-.

- **Stage 3:** The output of stage three is at the same time the output of the system. Thus, we can analyze the output of stage three both in terms of stage performance and global performance. The stage performance is pictured in Table 3, while the global performance analysis is deployed in Table 4 and will be the object of study in the next paragraphs. Table 3 shows that the SVM classifier yields to a reduction about 71% in the number of frames at the output, keeping the 75% of the contractions provided by stage two. Moreover, the imbalance rate of the final data set is reduced to 0.7.

Table 1: Performance analysis for the first stage of the cascade

	INPUT			OUTPUT							
Studio	Frames	Findings	Imb. Rate	Frames (%)		Findings (%)		Imb. Rate	Lost Findings (%)		Non-Cont. Frames (%)	
Video 1	29444	747	39.4	3192	10.84%	720	96.39%	4.4	27	3.61%	2472	77.44%
Video 2	28803	529	54.4	3027	10.51%	502	94.90%	6.0	27	5.10%	2525	83.42%
Video 3	27816	575	48.4	3185	11.45%	561	97.57%	5.7	14	2.43%	2624	82.39%
Video 4	38885	733	53.0	4025	10.35%	717	97.82%	5.6	16	2.18%	3308	82.19%
Video 5	17619	356	49.5	1849	10.49%	349	98.03%	5.3	7	1.97%	1500	81.12%
Video 6	27360	476	57.5	2943	10.76%	459	96.43%	6.4	17	3.57%	2484	84.40%
Video 7	27176	918	29.6	2903	10.68%	890	96.95%	3.3	28	3.05%	2013	69.34%
Video 8	12620	150	84.1	1366	10.82%	143	95.33%	9.6	7	4.67%	1223	89.53%
Video 9	25994	206	126.2	2953	11.36%	198	96.12%	14.9	8	3.88%	2755	93.29%
Video 10	27967	397	70.4	2948	10.54%	385	96.98%	7.7	12	3.02%	2563	86.94%
Avg:			61.3		10.78%		96.65%	6.9		3.35%		83.01%

Table 2: Performance analysis for the second stage of the cascade

	OUTPUT									
Studio	Frames (%)		Findings (%)		Imb. Rate	Lost Findings (%)		Non-Cont. Frames (%)		% Findings in Video
Video 1	2774	86.90%	697	96.81%	4.0	23	3.19%	2077	74.87%	93.31%
Video 2	2346	77.50%	474	94.42%	4.9	28	5.58%	1872	79.80%	89.60%
Video 3	2623	82.35%	548	97.68%	4.8	13	2.32%	2075	79.11%	95.30%
Video 4	3170	78.76%	673	93.86%	4.7	44	6.14%	2497	78.77%	91.81%
Video 5	1740	94.10%	341	97.71%	5.1	8	2.29%	1399	80.40%	95.79%
Video 6	2288	77.74%	453	98.69%	5.1	6	1.31%	1835	80.20%	95.17%
Video 7	2692	92.73%	869	97.64%	3.1	21	2.36%	1823	67.72%	94.66%
Video 8	804	58.86%	134	93.71%	6.0	9	6.29%	670	83.33%	89.33%
Video 9	678	22.96%	184	92.93%	3.7	14	7.07%	494	72.86%	89.32%
Video 10	1538	52.17%	363	94.29%	4.2	22	5.71%	1175	76.40%	91.44%
Avg:		72.41%		95.77%	4.6		4.23%		77.35%	92.57%

Table 3: Performance analysis for the third stage of the cascade

	OUTPUT									
Studio	Frames (%)		Findings (%)		Imb. Rate	Lost Findings (%)		Non-Cont. Frames (%)		% Findings in Video
Video 1	904	32.59%	595	85.37%	0.5	102	14.63%	309	34.18%	79.65%
Video 2	607	25.87%	343	72.36%	0.8	131	27.64%	264	43.49%	64.84%
Video 3	646	24.63%	405	73.91%	0.6	143	26.09%	241	37.31%	70.43%
Video 4	981	30.95%	547	81.28%	0.8	126	18.72%	434	44.24%	74.62%
Video 5	433	24.89%	266	78.01%	0.6	75	21.99%	167	38.57%	74.72%
Video 6	768	33.57%	339	74.83%	1.3	114	25.17%	429	55.86%	71.22%
Video 7	835	31.02%	603	69.39%	0.4	266	30.61%	232	27.78%	65.69%
Video 8	189	23.51%	111	82.84%	0.7	23	17.16%	78	41.27%	74.00%
Video 9	228	33.63%	122	66.30%	0.9	62	33.70%	106	46.49%	59.22%
Video 10	363	23.60%	248	68.32%	0.5	115	31.68%	115	31.68%	62.47%
Avg:		28.42%		75.26%	0.7		24.74%		40.09%	69.69%

Finally, the global performance of the system, viewing all the steps in the cascade as a whole black box, can be faced in multiple ways: From a clinical point of view, the experts are interested in assessing how many of the existing contractions our system is able to detect, namely, the system *sensitivity*, how many of the existing non-contractions our system is able to reject, namely, the system *specificity*, and finally, which the ratio between false contractions and real contractions at the output of the system is, i.e., the system *precision*. In addition to the latter, a ratio between the false contractions at the output of the system and the existing contractions in the video provides the expert with useful information (we define this quantity as false alarm rate, *FAR*). A rigorous definition of the former quantities in terms of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) can be stated in the following way:

Sensitivity	Specificity	FAR	Precision
$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$	$\frac{FP}{TP+FN}$	$\frac{TP}{TP+FP}$

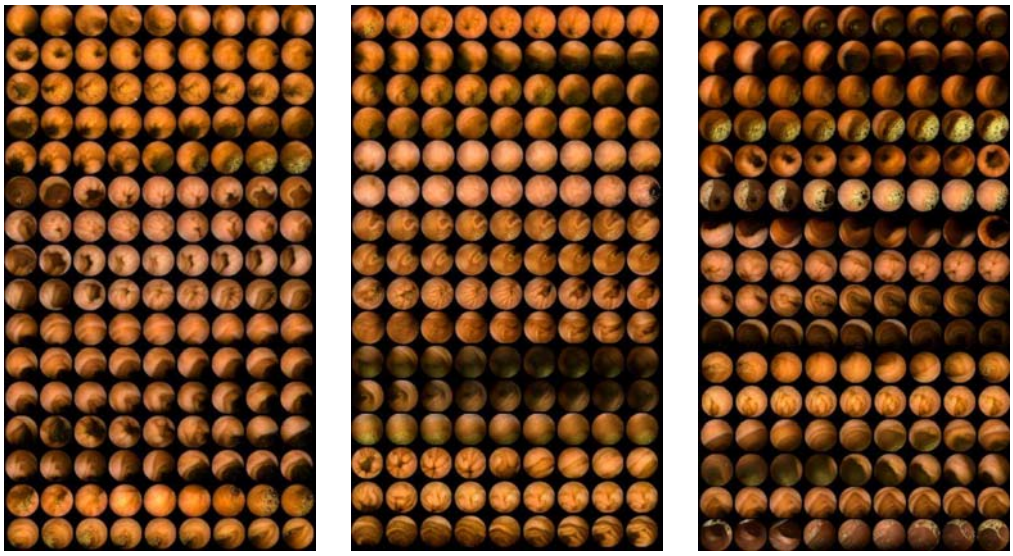
Table 4 summarizes the performance results of the cascade system: Our approach achieves an overall sensitivity of 69.68%, picking 80% for the studio referred as *Video 1*. The high overall specificity value of 99.59% is typical of imbalanced problems, and for this reason it is not generally useful for performance assessment tasks. However, FAR and precision carry out insightful information about what the output is like. The resulting precision value of 59.91% tells us that 6 out of 10 frames in the output correspond to true findings. FAR is similar, but in terms of noise (the bigger the FAR, the larger the number of false positives), and normalized by the number of existing contractions. For different videos providing an output with a fixed precision, those with the highest number of findings in video will have lower FAR. In this sense, A FAR value of one tells us that we have obtained as many false positives as existing

contractions in video. FAR and precision values are usually related, and Table 4 shows that the peeks of performance for both measures are found in the same two studios (*Video 6* and *Video 7*, outlined in bold type).

Table 4: Global system performance

Studio	Sensitivity		Specificity		FAR		Precision	
Video 1	595/747	**79.65%**	29135/29444	98.95%	309/747	41.37%	595/904	65.82%
Video 2	343/529	64.84%	28539/28803	99.01%	264/529	49.90%	343/607	56.51%
Video 3	405/575	70.44%	27575/27816	99.13%	241/575	41.91%	405/646	62.69%
Video 4	547/733	74.65%	38451/38885	98.88%	434/733	59.21%	547/981	55.76%
Video 5	266/356	74.72%	17452/17619	99.05%	167/356	46.91%	266/433	61.43%
Video 6	339/476	71.22%	26931/27360	**98.43%**	429/476	**90.13%**	339/768	**44.14%**
Video 7	603/918	65.69%	26944/27176	99.15%	232/918	**25.27%**	603/835	**72.22%**
Video 8	111/150	74.00%	12542/12620	99.38%	78/150	52.00%	111/189	58.73%
Video 9	122/206	**59.22%**	25888/25994	99.59%	106/206	51.45%	122/228	53.51%
Video 10	248/397	62.46%	27852/27967	**99.59%**	115/397	28.96%	248/363	68.32%
Avg:		**69.68%**		**99.12%**		**48.71%**		**59.91%**

In addition to the former numerical performance analysis, a more qualitative insight into the different sequences of positives and negatives provided by the system deserves to be accomplished. Figure 11 shows a set of paradigmatic examples for (a) detected contractions (true positives), (b) not detected contractions (false negatives), and (c) sequences which had not been previously labelled by the experts, but which our system classified as contractions (false positives). The detected contractions basically correspond to the paradigm of phasic contractions described in Section 2.2. In this sense, clear patterns of the intestinal lumen closing and opening are shown. It must be noticed that the presence of turbid liquid in some frames does not result in a rejection of this sequence by the turbid detector, because only the clearest turbid sequences are rejected. The not detected contractions share some common features: on the one hand, the open lumen is not always present at the beginning and the end of the sequence, both because the camera is not pointing towards the longitudinal direction of the gut, or because the selected contraction is spanning for more than the 9 frames -this could be likely linked to the blurring definition border between short tonic contractions and phasic contraction-. Moreover, the motion impression that the expert perceives during the video visualization is not present in the deployed sequence of frames. In this sense, we performed some tests consisting of showing the experts a set of paradigmatic sequences containing doubtful contractions both by visualizing them in the video at a visualization ratio of 3 frames per second, and showing the same sequences deploying the 9 frames as in Figure 11. We found that the experts usually labelled a higher number of contractions during the video visualization than looking at the deployed sequence. This fact drives us to think that the motility characterization should be performed in a more subtle detail, in order to detect the apparently slight changes in some sequences shown in Figure 11(b), but which actually seem to be clear for the expert during the visualization process. Finally, the false positives analysis supply very interesting results: On the one hand,



(a) (b) (c)

Figure 11: Some example sequences provided by the system. (a) Correctly detected contractions. (b) Non-detected contractions (false negatives). (c) Sequences which had not been labelled by the experts, but detected as contractions (false positives).

our system shows its ability to detect real contractions which the experts did not get to label -an example of these sequences is rendered in the fifth row of Figure 11(c). This is a reasonable result, since one of the main drawbacks associated with motility assessment by manual labelling is the growing stress and fatigue which takes place during visualization, yielding to a loss of effectiveness in the final outcome. A rough study over the false positives of the ten analyzed videos showed that about the 10% of the false positives consisted of this kind of sequences. On the other hand, the sequences shown in Figure 11(c) display the inherent difficulty related to the high variability of patterns present at the output of the system: The lateral movement of the camera while focusing the lumen which can be confused with the pattern of its contraction, the differences in illumination creating shadows which can be confused with the lumen, the multiple patterns of wrinkles which can provide a high response to the lumen detector, and the residual presence of patterns of turbid liquid, share the main responsibility in the false positives. We suggest that many of these problems may be tackled by a deeper study about the textural information provided by the lumen, both in the relaxed stage and the contraction activity. This issue, and some other proposed approaches, are deeply deployed in Section 6.

4.2 Validation of the system operation point

Providing that the set of parameters $P^0 = \{P_1^0, P_2^0, P_3^0, P_4^0\}$, was obtained in an exhaustive heuristical search, we must assess that P^0 does correspond with an optimal operation point, in terms of system performance. In order to assess this issue, we proceeded with a forward-propagation algorithm for parameter selection, which is deployed in detail in Appendix A. The procedure used essentially matches the following highlights: We reset all the parameters to P^0, and established a range of possible values for each of them: 16 values for P_1 within the interval [0:15], 11 values for P_2 within the interval [10:210], 11 values for P_3 within the interval [500:1000], and the 8 heuristically selected values for P_4 [0.001,0.005,0.010,0.030,0.050,0.100,0.500,1.000]. The choice of these intervals was performed based on the minimum and maximum values for each stage. The interval of the last parameter (the SVM γ) was carefully selected based on the observation of a substantial variation of the classifier performance. After the initialization step, the system was evaluated for all the possible values of P_1 within the defined range, and the best operation point (P_1^{Best}) was selected according to the performance criteria defined in Appendix A. The value of P_1^0 was substituted by P_1^{Best}, repeating the same procedure for the rest of the parameters in a sequential way (P_2, P_3 and P_4). The whole procedure was repeated 5 runs and the final set P^{Best} was obtained by averaging $P^{Best_{i,i=1:5}}$.

Both the fast forward algorithm, and the performance criteria chosen are justified by the following reasons: On the one hand it must be taken into account that each single parameter modification has impact on the frames which are to be filtered by each specific stage, not only in the final assessment test, but also in the videos which are used for training in the leave-one-out strategy. In other words, when we vary one parameter, we must re-run all the system for each one of the 9 videos used for training, and we must apply a new leave-one-out strategy for each of them, training their classifiers using the 8 remaining videos. This clearly appears not to be computationally affordable using another parameter selection strategy which would imply a substantial increase in its computation time. On the other hand, a performance criterion function based on the global classification error does not appear to be a reliable metric in this context. In order to tackle the issue of performance assessment in imbalanced problems, several authors have proposed different solutions, including the use of the g-metric, the F-metric, and others [21]. Among the clinical community, the use of a trade-off between sensitivity and some other measure is widely extended. For our case, we demanded our experts to provide us with the reference of the performance threshold which should be used in the trad-off function, arriving at a final compromise with $Sensitivity \geq 70\%$. Finally, we implemented this criterion within the criteria function defined in Appendix A.

In order to accomplish a graphical analysis of this procedure, let us fix our attention on the ROC curves plots shown if Figure 12. In ROC curves, both sensitivity and specificity are plotted (properly speaking, the *fp-ratio* is plotted, defined as 1-specificity -notice the difference in the axes scale-) rendering the possible operation points of the system, and constituting a helpful tool for performance analysis. Figure 12 plots the points of the ROC curves segments corresponding to the different operation points provided by the different values of the parameter vector P after 5 runs. Each run is represented with a different symbol and color. Each graph (a), (b), (c) and (d) corresponds with one parameter in P (P_1, P_2, P_3 and P_4). Figure 13 shows the points of the same ROC curves segments clustered by the same parameter. In these plots, each operation point is centered in the mean value of sensitivity and fp-ratio after the 5 runs, and the length of the ellipses axes is proportional to its standard deviation. ROC curves in Figure 12 show that our system appears to be robust, in the sense that the trade off between sensitivity and specificity is kept for each run. The less fp-ratio, the less sensitivity is achieved. Furthermore, our system shows to be stable, in the sense that for several runs, the resulting operation point is confined in the ellipses drawn in the plots rendered in Figure 13, showing no hysterical responses. We can observe the monotonically growing curves for the different parameter values, and the global displacement of the curve segment from the 60% to the 70% of sensitivity - (a) to (c)-. The parameter P_4 (γ value of SVM) presents the widest range of variability, being consistent with the role of γ, which controls the margins which directly affect the support vectors used for classification.

The final performance of the system was calculated in two different ways: 1) averaging the performance point of the 5 runs of the validation procedure tuned with $P^{Best_{i,i=1:5}}$, and 2) averaging 5 runs of the system tuned in P^{Best}. Table 5 shows these results in comparison with the performance of the system tuned to P^0, exposed in the previous subsection. The final outcome confirms our hypothesis over P^0, since the confidence intervals of the performance values for the heuristically obtained parameters and those provided by the forward-propagation algorithm overlap both for sensitivity and FAR, assessing the equivalence of P^0 and P^{Best} in terms of performance.

Table 5: Performance operation point for the different parameters

Parameter	Sens.(std)		FAR(std)	
P^0	68.88	(0.51)	46.96	(0.79)
$P^{Best_{i,i=1:5}}$	69.35	(1.10)	47.96	(1.58)
P^{Best}	69.68	(0.44)	48.72	(0.54)

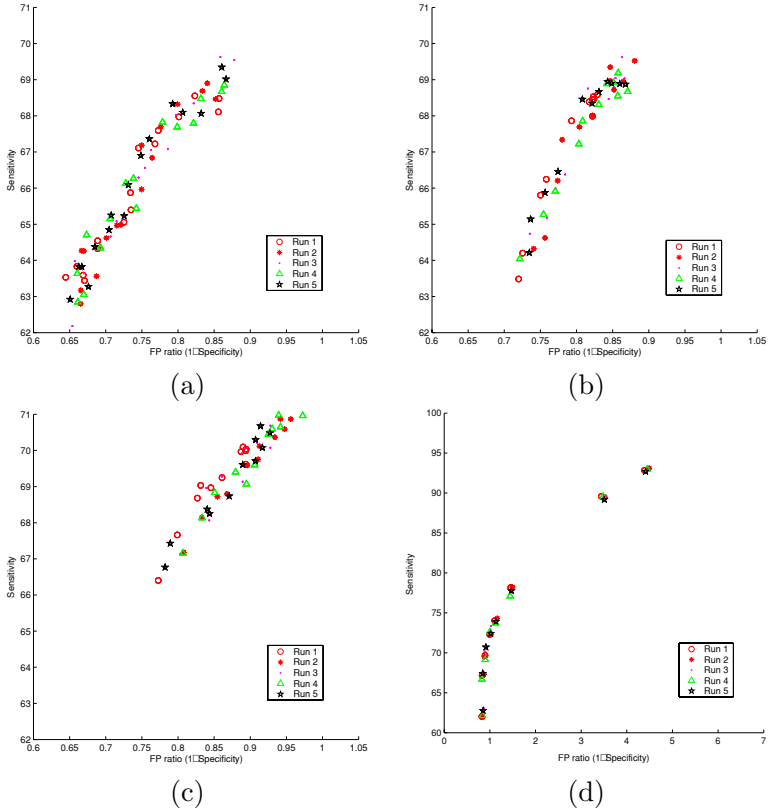


Figure 12: ROC curves segments for the forward parameter selection procedure for (a) P_1, (b) P_2, (c) P_3 and (d) P_1. Each symbol represents each of the different 5 runs. The different points of each symbol represent the different performance pairs of sensitivity vs. fp-ratio.

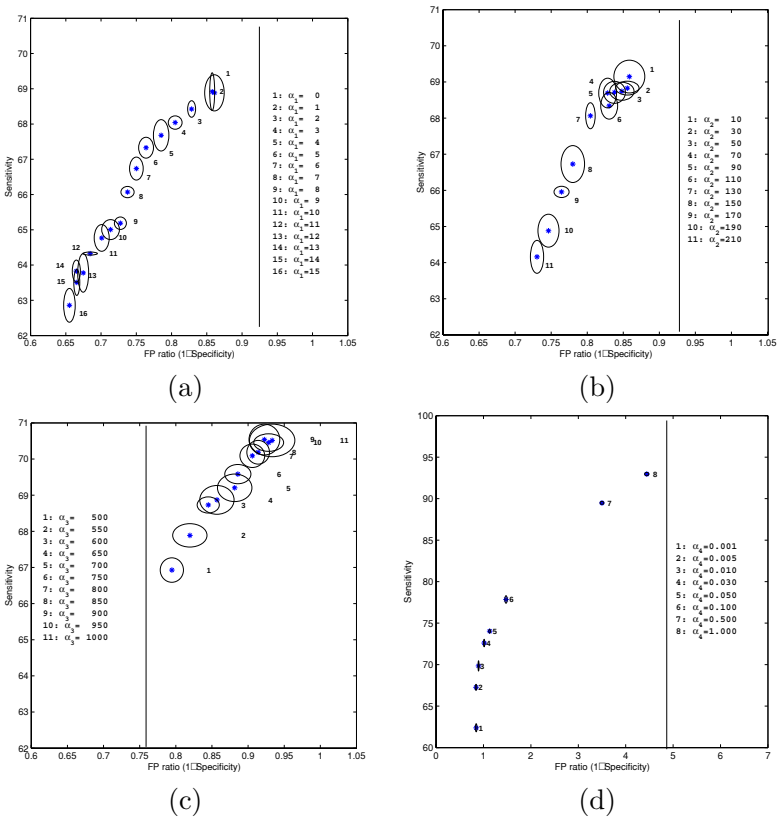


Figure 13: ROC curves segments for the forward parameter selection procedure for (a) P_1, (b) P_2, (c) P_3 and (d) P_1. The points in Figure 12 are grouped by its parameter value, instead of runs. The mean of each ellipse represent the mean of the performance pair obtained for that parameter after 5 runs. The exes of the ellipses are related to the resulting mean variance.

5 Conclusion

This work addressed the problem of the automatic detection of intestinal contractions in capsule video endoscopy, a novel and highly challenging issue in medical imaging. The main novelty of our contribution is based on tackling the assessment of intestinal motility with a machine learning approach, which joins both classical image processing techniques and the use of diverse strategies for facing the low prevalence of contractions in video. The main outcome is that we turned a useful but not feasible clinical routine, such as the manual labelling of intestinal contractions in video endoscopy studios, into a feasible clinical routine by means of their automatic detection, obtaining reasonable performance results.

We showed the design of the system ,in terms of sequential stages, to be helpful from a two-fold perspective: On the one hand, this approach lets the experts identify different features related to intestinal motility in capsule video endoscopy, such as the presence of high content of intestinal juices which hinders the video visualization, or the detection of spans of time with no motility activity. By using this modular perspective, domain knowledge can be easily added to the system by the experts, by means of the inclusion of new sequential stages to the cascade. On the other hand, we showed the rejection of negatives in a sequential way to be a useful strategy for dealing with the skewed distribution of positives -i.e., contractions- and negatives -i.e., non-contractions- along the video data. We provided a detailed explanation and study of the different steps that we defined in the cascade, showing intermediate measures of performance for each stage. In addition to this, a general validation study of the different parameters used in the cascade was deployed. Finally, we provided the global performance point of our system both in a qualitative and a quantitative way, by means of usual performance measures such as sensitivity, specificity and precision, and by introducing our own false alarm rate (FAR) as a useful metric for the specialists.

6 Future work

Several challenging issues are part of our current and future lines of work: 1) The search of optimal descriptors for the intestinal contractions sequences in capsule video endoscopy appears as an open fieldwork. The main approaches we are dealing with include more sophisticated textural descriptors. In this sense, we suggest that the inclusion of information regarding the wrinkle pattern which generally appears associated with the contracted lumen might be studied with deeper attention. 2) The use of applied techniques to handle the skewed distributions of the data sets should deserve, from our point of view, a special consideration as well. Although

under-sampling together with SVM appears to provide a reasonable performance in our data set, we believe that an improvement in the classification results can be achieved, both automatically identifying wide regions of the feature space associated with non-contractions, and improving the classifier performance by means of ensemble methodologies [26]. 3) Finally, the performance assessment of this kind of clinical scenarios present inherent problems due to the lack of viability in labelling all the video frames in a explicit way as positives or negatives. We are carrying out research in order to develop interactive tools which could be used by the experts in order to provide useful feedback information which could help us to detect specific patterns of motility. All these strategies, together with the extension of our system to the analysis of tonic contractions and postprandial patients, are part of current experiments that our group is developing in close contact with our group of specialists. We are to present the impact of this novel methodology in terms of full clinical assessment and the evolution of the lines of work described above for future pieces of research.

A Fast-forward parameter selection algorithm

BEGIN
SET ranges for parameters:
$R_1 \rightarrow [0:15]$ {16 values}
$R_2 \rightarrow [10:210]$ {11 values}
$R_3 \rightarrow [500:1000]$ { 11 values}
$R_4 \rightarrow [0.001, 0.005, 0.010, 0.030, 0.050, 0.100, 0.500, 1.000]$ {8 values}
for $i = 1$ to 5 **do**
  Set parameters: $P = P^0 = \{P_1^0 = 0, P_2^0 = 50, P_3^0 = 650, P_4^0 = 0.01\}$ {initialization}
  for $j = 1$ to 4 **do**
    Calculate the system performance substituting P_j^0 with each value of R_j.
    Apply the Performance Criteria to obtain P_j^{Best}.
    Substitute P_j^0 with P_j^{Best} in P.
  end for
  $P^{Best_i} = \{P_1^{Best}, P_2^{Best}, P_3^{Best}, P_4^{Best}\}$
end for
$P^{Best} = avg(P^{Best_i})$
END

For all the performance pairs (Sensitivity, FAR) obtained for each parameter:
if For all the pairs, Sensitivity ≥ 70 **then**
  We chose the parameter that achieves the higher sensitivity.
else
  We select the two parameters with a closest value to 70(higher or lower)
  We choose the parameter which minimizes the error function:
  $sens * (a(sens)^2 + b(FAR)^2), a, b = 1$
end if

References

[1] J. Kellow, M. Delvaux, F. Aspriroz, *et al.*, "Principles of applied neurogastroenterology: physiology motility-sensation," *Gut*, vol. 45, no. 2, pp. 1117–1124, 1999.

[2] E. M. Quigley, "Gastric and small intestinal motility in health and disease," *Gastroenterology Clinics of North America*, vol. 25, pp. 113–145, 1996.

[3] ——, "Disturbances in small bowel motility," *Baillieres Best practice and research. Clinical gastroenterology*, vol. 13, no. 3, pp. 385–395, 1999.

[4] M. B. Hansen, "Small intestinal manometry," *Physiological Research*, vol. 51, pp. 541–556, 2002.

[5] M. P. Tjoa and S. M. Krishnan, "Feature extraction for the analysis of colon status from the endoscopic images," *Biomedical Engineering OnLine*, vol. 2, pp. 3–17, 2003.

[6] S. A. Karkanis, D. K. Iakovidis, *et al.*, "Computer aided tumor detection in endoscopic video using color wavelet features," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, pp. 141–152, 2003.

[7] G. Magoulas, V. Plagianakos, *et al.*, "Neural network-based colonoscopic diagnosis using online learning and differential evolution," *Applied Soft Computing*, vol. 4, pp. 369–379, 2004.

[8] M. M. Zheng, S. M. Krishnan, and P. Tjoa, "A fusion-based clinical support for disease diagnosis from endoscopic images," *Computers in Biology and Medicine , Article in Press.*

[9] V. S. Kodogiannis and H. S. Chowdrey, "Multi-network classification scheme for computer-aided diagnosis in clinical endoscopy," *Proceedings of the International Conference on Medical Signal Processing (MEDISP)*, pp. 262–267, 2004.

[10] M. Boulougoura, V. Wadge, *et al.*, "Intelligent systems for computer-assisted clinical endoscopic image analysis," *Proceedings of the 2nd IASTED Conference on Biomedical Engineering Innsbruck*, pp. 405–408, 2005.

[11] G. Iddan, G. Meron, *et al.*, "Wireless capsule endoscopy," *Nature*, vol. 405, p. 417, 2000.

[12] N. V. Chawla, "Data duplication: an imbalanced problem?" in *Workshop on Learning from Imbalanced Datasets II, ICML*, 2003.

[13] T. Fawcett and F. J. Provost, "Adaptive fraud detection," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 291–316, 1997.

[14] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Mach. Learn.*, vol. 30, no. 2-3, pp. 195–215, 1998.

[15] P. Domingos, "Metacost: A genera method for making classifiers cost-sensitive," in *Procdeedongs of the Fith ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155–164.

[16] M. C. Monard and G. E. Batista, "Learning with skewed class distribution," in *Advances in Logic, Artificial Intelligence and Robotics*, 2002, pp. 173–180.

[17] B. Zadrozny and C. Elkan, "Learning and making decisions when costs and probabilities are both unknown," in *Proceedings of the 7th ICKDDM*, 2001.

[18] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[19] J. C. Russ, *The Image Processing Handbook*. CRC Press. 2nd Edition, 1994.

[20] I. Guyon, J. Weston, *et al.*, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 389–422, 2002.

[21] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets." in *ECML*, 2004, pp. 39–50.

[22] F. Vilarino, P. Spyridonos, *et al.*, "Experiments with svm and stratified sampling with an imbalanced problem: Detection of intestinal contractions," *LNCS.*, vol. 3687, no. 2, pp. 783–791, 2005.

[23] S.Theodoridis and K.Koutroumbas, *Pattern Recognition*. Elsevier, 20003.

[24] T. Ojala, M. Pietik, *et al.*, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, 2002.

[25] (2005) Given imaging, ltd. [Online]. Available: http://www.givenimaging.com/

[26] F. Vilarino, L. Kuncheva, *et al.*, "Roc curves and video analysis optimization in intestinal capsule endoscopy," *Pattern Recognition Letters. In press*, 2005.

Video-based face processing: 2D and 3D approaches

José Miguel Buenaposada, Enrique Muñoz, Luis Baumela
Perception for Computers and Robots Group (PCR),
Universidad Politécnica de Madrid (UPM), Facultad de Informática,
Campus de Montegancedo s/n, E-28660 Boadilla del Monte, Madrid, Spain
http://www.dia.fi.upm.es/~pcr

Abstract

Being able to interpret facial expressions is essential in order to understand and be understood by others. Computers need facial interpretation capabilities if they are to interact with humans in a natural way. In recent years Computer Vision has emerged as the most promising technology to achieve such a challenging goal. There are various ways to perform facial expression analysis in video sequences using Computer Vision, which can be broadly grouped into 2D and 3D approaches. In this paper we describe the 2D and 3D facial expression analysis algorithms that are being developed within the PCR group at the UPM.

Keywords: Facial Expressions Recognition, Face processing on Video, Face tracking, Computer Vision.

1 Introduction

Nowadays facial animation is attracting the interest of the computer industry more than ever before. There are several reasons for such interest, but the main one is the wide range of possible applications: advanced human-computer interfaces, interactive computer games, virtual reality systems, etc.

Facial animation has emerged as a challenging task for the Computer Vision, Computer Graphics and Voice Processing research communities. The main goal of the Computer Vision techniques is to estimate the human face pose and to quantify the deformation of its non-rigid parts. Computer Graphics develop procedures to render photo-realistic images. Finally, advanced Voice Processing algorithms provide realistic sound to the animations. The confluence of these three research areas could, for example, make possible in a not so far future to create a new film of Marylin Monroe with her original voice and appearance by processing old footage.

Edited by F.Pla, P.Radeva, J.Vitrià, 2006.

In this paper we will present some solutions that we have developed for the human face analysis[1] problem. From a Computer Vision point of view, tracking a face is a challenging task because the human head, due to the articulation of the jaw and the deformation of skin, is a non-rigid object composed mostly of low textured regions. State-of-the-art commercial systems for motion capture overcome this problem by introducing artificial markers (see, for example, http://www.vicon.com).

The marker-less face analysis problem is still an open research issue. To solve it we use a priori information, which is represented in our system using a model of the target face to be tracked. The main obstacle to build such a model is the great variability of the human face. This variability comes from the anatomical differences between people and from other factors which influence the facial appearance of a person: the point of view, the illumination, facial expressions, and finally, the presence of glasses, hats, etc.

The isolation and description of all these sources of variation is the main goal of a face analysis system. The challenge in building such a system consists of finding a model general enough to represent any source of variation and with enough precision to describe an image of the face.

This problem has been tackled from two different approaches:

- *Feature based tracking.* It consists of tracking a discrete set of texture elements on the image, such as the corners of the eyes, nose and mouth, or the contours of the mouth, eyes, etc. The motion parameters of each of these elements give us information about the motion of the whole face and that of its non-rigid elements.

- *Analysis-Synthesis procedures.* A dense generative face model is used to create a parametrised image of the target face. Tracking is done by minimising the differences between the analysed image and the one generated using the model.

Feature-based methods were the first to be introduced [3, 13], since they had already been successfully used during the mid nineties to solve the structure from motion problem. These techniques could only estimate the motion of textured regions, therefore, they only provided sparse information about the deformation of the face. They were unable to estimate subtle expressions in low textured face regions.

Analysis-synthesis techniques solve the problem globally, in such a way that the motion of textured regions help to disambiguate that of low textured parts. They are usually based on advanced computer graphics models which render photo-realistic images of the face. Let I be an image of the target that is to be tracked, and let

[1] We assume face analysis, face motion capture and face tracking are synonymous terms.

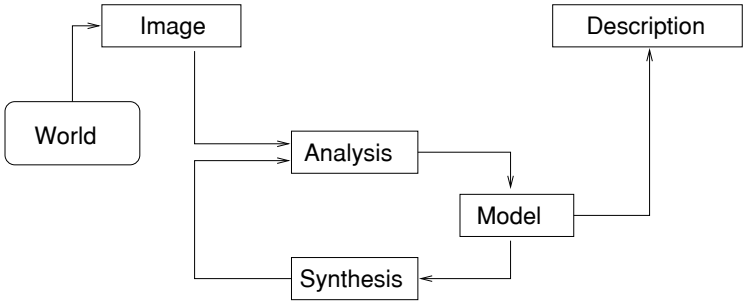


Figure 1: Analysis-Synthesis work-flow. A realistic image is rendered using a graphical model and compared with the image to be analysed. The model parameters' are refined iteratively by making the rendered image appear like the real image. The outcome of this analysis is a description expressed in terms of the model parameters that minimise the differences between both images.

$G(\mathbf{p})$ be the image rendered with a model with parameters $\mathbf{p}$. The analysis of the face consists of finding the model parameters, $\mathbf{p}$, that render an image most similar to the appearance of the target (see Fig. 1). This is achieved by solving the following minimisation problem:

$$\arg\min_{p} ||I - G(p)||^2. \tag{1}$$

Some of the constituents of model G may be analytically derived from the laws of physics (for example, rigid motion and in some cases illumination). Other components are difficult to represent analytically and they are usually learnt from examples, by means of statistical procedures. These are face identity, facial expressions and illumination.

Early solutions within this approach modelled the face as a rigid 3D textured object and tracked it by using a model of face texture mapped onto planar [14], ellipsoidal [4] or cylindrical [17] 3D models. Later, changes in facial expressions were also modelled by using linear subspace representations of face appearance [11], or linear models of shape+texture such as the 2D *Active Appearance Models* (AAMs) [10] or the 3D *Morphable Models* (MMs) [6]. More recently, non-linear subspaces have also been used to model the appearance of a face across changes in pose, facial expression and illumination [18].

The major drawback of the analysis-synthesis approach is the difficulty of building such realistic models. Another important issue is devising efficient minimisation procedures to solve (1). Standard optimisation techniques have to be customised in order to efficiently deal with the amount of data involved (thousands of points are normally used) and the non-convexity of the cost function.

The problem of efficiency is considered in section 2. In section 3 we introduce the *principal component analysis*, a mathematical procedure, quite popular in the Computer Vision community, to learn models of the face. In sections 4 and 5 we introduce the 2D and 3D approaches for face tracking and present some of our results. Finally, in section 6, we draw some conclusions and discuss futures venues for research.

2 Efficient minimisation

One interesting side of the problem, especially if we want to achieve real-time performance while tracking, is the development of efficient minimisation procedures for (1), or its particularisation for appearance (6) and shape+appearance models (10).

In general, $G(\mathbf{p})$ establishes a nonlinear relation between the model parameters' $\mathbf{p}$ and the grey levels of the rendered image. Therefore, the cost function to minimise (1) is non-convex, with many local minima. A common approach is to start from a known solution for the first image (we obtain this solution by employing much simpler tracking algorithms) and incrementally compute the parameters of all other images in the sequence:

$$\delta\boldsymbol{\mu}(t) = \arg\min_{\delta\boldsymbol{\mu}} ||I(t+\delta t) - G(\mathbf{p}(t)+\delta\boldsymbol{\mu})||^2. \tag{2}$$

In this case, we can make a Taylor series expansion of (2) to get a linear approximation at time instant t:

$$\delta\boldsymbol{\mu}(t) = \arg\min_{\delta\boldsymbol{\mu}} ||I(t+\delta t) - G(\mathbf{p}(t)) - \mathtt{J}(\mathbf{p}(t))\delta\boldsymbol{\mu}||^2, \tag{3}$$

where

$$\mathtt{J}(\mathbf{p}(t)) = \left.\frac{\partial G(\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}(t)}$$

is the Jacobian of the grey levels of the rendered image with respect to the model parameters'. It represents our *a priori* information about the target. That is, how the grey levels (or colours) of the image change as the model parameters' vary.

The minimisation in (3) can be trivially solved by least squares:

$$\delta\boldsymbol{\mu}(t) = (\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top\mathcal{E},$$

where $\mathcal{E} = I(t+\delta t) - G(\mathbf{p}(t))$ is the error made when comparing the image acquired at time instant $t+\delta t$ with the rendered image using the model parameters at time instant t. Whenever the $\mathtt{J}^\top\mathtt{J}$ matrix is singular, the object motion can not be estimated.

This happens when the object texture does not give enough information to estimate the motion. This is a generalisation of what is known in Computer Vision as the *aperture problem* for the estimation of optical flow.

The main obstacle to solve (3) in real-time is the computation of the Jacobian matrix. $\mathtt{J}(\mathbf{p}(t))$ is a matrix that has one row for each pixel in I and as many columns as parameters in $\mathbf{p}$, and it has to be recomputed for each image in the sequence, since it depends on $\mathbf{p}(t)$. Two efficient solutions have been introduced in the Computer Vision literature, which can be used to develop tracking algorithms with real-time performance: the *Jacobian Factorisation* approach [14] and the *Inverse Compositional Algorithm* [1]. The efficiency of the first algorithm is achieved by factoring the Jacobian into the product of a constant matrix, which depends on the image texture, and a small matrix, $\Sigma(p(t)))$, that depends on the motion parameters at time instant t, which can be computed in real-time. The Inverse Compositional Algorithm roots its efficiency in employing a constant Jacobian computed at a fixed reference image. Both solutions have its limitations. The Jacobian factorisation has been obtained for planar appearance models with affine or projective rigid deformations [8, 9]. But the application of this procedure to the shape+appearance models still remains to be investigated. On the other hand, using some approximations, the Inverse Compositional Algorithm has been applied to the real-time tracking of Active Appearance Models (AAMs) [19].

3 Principal Component Analysis

Principal Component Analysis (PCA) is a basic tool in the construction of generative linear models, either appearance or shape+appearance-based ones. It is used to learn the model components which cannot be described analytically.

Let $\mathcal{X} = \{\mathbf{x}_i,\ i = 1 \ldots N\}$ be a set of points of dimension $\dim(\mathbf{x})=d$ (see Fig. 2) and let $\mathtt{C}_{d\times d} = E\{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top\}$ be the data covariance matrix. The eigenvectors $\mathbf{u}_j$ of $\mathtt{C}$, $\mathtt{C}\mathbf{u} = \lambda\mathbf{u}$, define an orthogonal basis for the data set, aligned with the maximum variability axes of the data. Now, any $\mathbf{x}_i$ can be expressed as

$$\mathbf{x}_i = \bar{\mathbf{x}} + \sum_{k=1}^{d} c_{ik}\mathbf{u}_k = \mathtt{B}\mathbf{c}_i, \tag{4}$$

where c_{ik} are the coordinates on the new basis, $\mathtt{B} = [\bar{\mathbf{x}}, \mathbf{u}_1, \ldots, \mathbf{u}_d]$ and $\mathbf{c}_i = [1, c_{1k}, \ldots, c_{dk}]^\top$.

The best approximation of dimension m, $m < d$, to the original data set is, in a least-squares sense, the result of discarding the terms depending on the coordinates associated to the minimum variability axes in equation (4), that is, those terms depending on the $\mathbf{u}_k$ associated to the $d - m$ minimum eigenvalues. This property

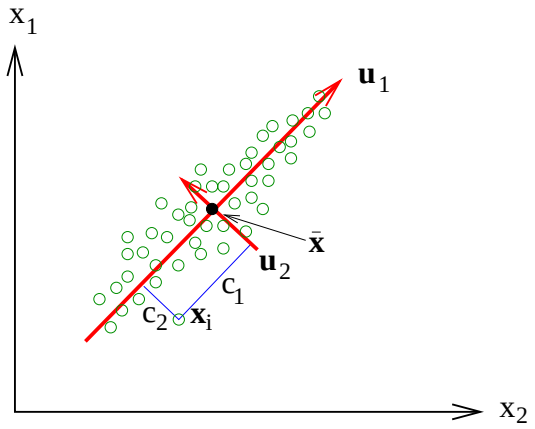


Figure 2: Principal Component Analysis. Maximum variability axis.

will be used later in the paper to model face appearance variations using a small number of parameters.

4 Appearance-based models

Here an object is represented by a set of grey levels (an image) on a planar surface (the face). The rigid transformations of the object are modelled by texture warping operations on the image plane. The deformation of the face is learnt from examples using PCA. To achieve this goal, an image I of dimensions $r \times s$ can be seen as a point in an $r \times s$-dimensional space (see Fig. 3). If we represent in this space the images

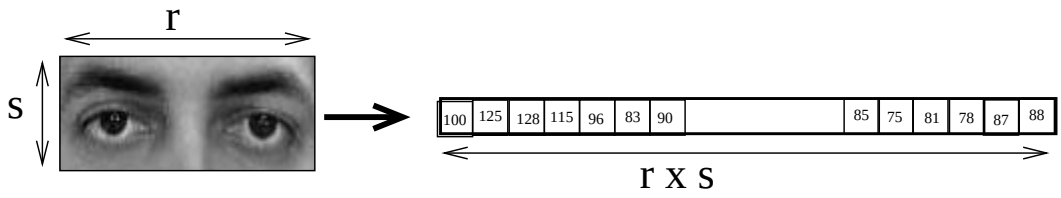


Figure 3: An image is a point in an space of $r \times s$ dimensions.

of any possible deformation of the object (the appearance space), the corresponding $r \times s$ dimensional points will lay on a subspace that may be modelled linearly using a basis of $m \ll r \times s$ dimensions (see Fig. 4).

The tracking algorithm presented in this section can be seen as an extension of the Hager and Belhumeur's *Jacobian factorisation* [14] in which we impose no

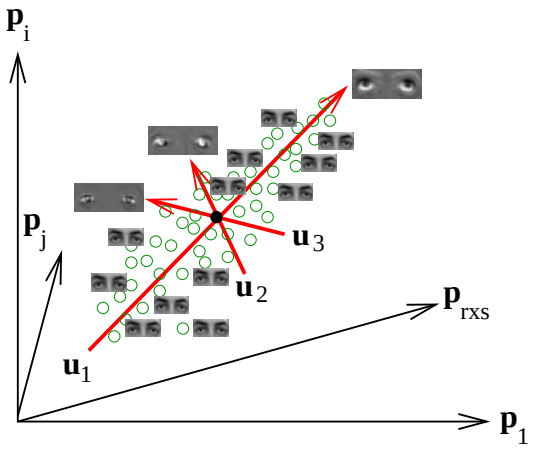


Figure 4: Appearance space for the eyes.

restrictions on the linear subspace model used. It is also related to the Black and Jepson's *Eigentracking* [5], but instead of computing the motion parameters by using a gradient descent procedure in which the target image Jacobian must be computed for each frame in the sequence, we use a set of precomputed motion templates which alleviate the computations that have to be performed on line.

Let P be the pixels in an image that belong to a target. The subspace constancy equation holds for all pixels in P

$$I(f(\mathbf{x}, \boldsymbol{\mu}(t)), t) = [\mathtt{B}\mathbf{c}(t)](\mathbf{x}) \quad \forall x \in P, \tag{5}$$

where $\mathbf{x}$ is the vector of co-ordinates of a point in image I, $\mathtt{B}$ is the subspace base matrix, $\mathbf{c}$ is the vector of subspace coefficients, and $I(f(\mathbf{x}, \boldsymbol{\mu}), t)$ is the image acquired at time t rectified with motion model $f(\mathbf{x}, \boldsymbol{\mu})$ and motion parameters $\boldsymbol{\mu}$. By $[\mathtt{B}\mathbf{c}](x)$ we denote the value of $\mathtt{B}\mathbf{c}$ for the pixel with position $\mathbf{x}$ in the image. Matrix $\mathtt{B}$ is of dimension $N \times k$, where N is the number of pixels per image and k is the number of basis vectors in the subspace. Intuitively (5) states that the rigidly rectified image $I(f(\mathbf{x}, \boldsymbol{\mu}), t)$ can be expressed as a linear combination of the appearance subspace basis vectors, $\mathtt{B}$.

In this case, equation (1) can be rewritten as

$$\min_{\boldsymbol{\mu}, \mathbf{c}} E(\boldsymbol{\mu}, \mathbf{c}) = \min_{\boldsymbol{\mu}, \mathbf{c}} ||\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}(t)), t) - \mathtt{B}\mathbf{c}(t)||^2, \tag{6}$$

where $\mathbf{I}(\mathbf{x})$ is $I(\mathbf{x})$ in vector form (e.g. scanning I by rows or columns). The parameters $\boldsymbol{\mu}(t)$ and $\mathbf{c}(t)$ will model respectively, position and facial expression at time instant t.

Tracking consists of estimating for each image in the sequence the values of the motion, $\boldsymbol{\mu}$, and appearance, $\mathbf{c}$, parameters. Here, the analytic component of G, $f(\mathbf{x}, \boldsymbol{\mu}(t))$, has been moved to $I(\mathbf{x}, t)$, for convenience.

In order to solve (6), a Taylor series expansion of $\mathbf{I}$ at $(\boldsymbol{\mu}(t), t)$ is made, producing a new error function

$$E(\delta\boldsymbol{\mu}, \mathbf{c}) = ||\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}(t)+\delta t), t+\delta t) - \mathtt{B}\mathbf{c}(t+\delta t)||^2 \approx ||\mathtt{M}\delta\boldsymbol{\mu} + \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}(t)), t+\delta t) - \mathtt{B}\mathbf{c}(t+\delta t)||^2, \tag{7}$$

where $\mathtt{M} = \frac{\partial \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}))}{\partial \boldsymbol{\mu}}$ is the $N \times n$ ($n = \dim(\boldsymbol{\mu})$) Jacobian matrix of I.

In the following subsections we will introduce a procedure for precomputing a set of motion templates which efficiently minimise (7) for any linear subspace model.

4.1 Jacobian matrix factorisation

One of the obstacles for minimising (7) on line, while tracking, is the computational cost of estimating $\mathtt{M}$ for each frame. Following an approach similar to [14], $\mathtt{M}$ can be expressed in terms of the gradient of the subspace basis vectors, $\mathtt{B}_\nabla$, which are constant, and the motion and appearance parameters $(\boldsymbol{\mu}, \mathbf{c})$, which vary over time. If we choose a motion model f such that $\mathtt{C} f_{\mathbf{x}}(\mathbf{x}_i, \boldsymbol{\mu})^{-1} f_{\boldsymbol{\mu}}(\mathbf{x}_i, \boldsymbol{\mu}) = \Gamma(\mathbf{x}_i)\boldsymbol{\Sigma}(\boldsymbol{\mu}, \mathbf{c})$ then $\mathtt{M}$ can be factored into [2]

$$\mathtt{M}(\boldsymbol{\mu}, \mathbf{c}) = \begin{bmatrix} \mathtt{B}_\nabla(\mathbf{x}_1)\Gamma(\mathbf{x}_1) \\ \vdots \\ \mathtt{B}_\nabla(\mathbf{x}_N)\Gamma(\mathbf{x}_N) \end{bmatrix} \Sigma(\boldsymbol{\mu}, \mathbf{c}) = \mathtt{M}_0 \Sigma(\boldsymbol{\mu}, \mathbf{c}),$$

where $\mathtt{B}_\nabla(\mathbf{x}_i)$ is the Jacobian of $\mathtt{B}$ with respect to the image co-ordinates. Then $\mathtt{M}_0$ is a constant matrix and Σ depends on $\mathbf{c}$ and $\boldsymbol{\mu}$.

4.2 Minimising $E(\boldsymbol{\mu}, \mathbf{c})$.

As $\mathtt{M}$ depends on both, $\boldsymbol{\mu}$ and $\mathbf{c}$, (7) defines a nonlinear cost function over $\delta\boldsymbol{\mu}$ and $\mathbf{c}$. The optimisation algorithm that we use first assumes $\mathbf{c}$ constant and computes the minimum of $E(\boldsymbol{\mu}, \mathbf{c})$ w.r.t. $\boldsymbol{\mu}$,

$$\delta\boldsymbol{\mu} = -(\Sigma^\top \mathcal{M} \Sigma)^{-1} \Sigma^\top \mathtt{M}_0^\top [\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}(t)), t + \delta t) - \mathtt{B}\mathbf{c}(t)], \tag{8}$$

where $\mathcal{M} = \mathtt{M}_0^\top \mathtt{M}_0$. The new motion parameters at time instant $t + \delta t$ will be $\boldsymbol{\mu}(t + \delta t) = \boldsymbol{\mu}(t) + \delta\boldsymbol{\mu}$. Then we minimise E for $\mathbf{c}$, assuming $\boldsymbol{\mu}$ constant,

$$\mathbf{c}(t + \delta t) = \mathtt{B}^\top [\mathtt{M}\delta\boldsymbol{\mu} + \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}(t)), t + \delta t)].$$

[2] $f_{\mathbf{z}}$ is the derivative of f w.r.t. the $\mathbf{z}$ derivative

Once we have $\mathbf{c}$, we can refine the estimation of $\delta\boldsymbol{\mu}$ by using (8) again. Normally two or three iterations are enough to reach a stable solution.

4.3 Results.

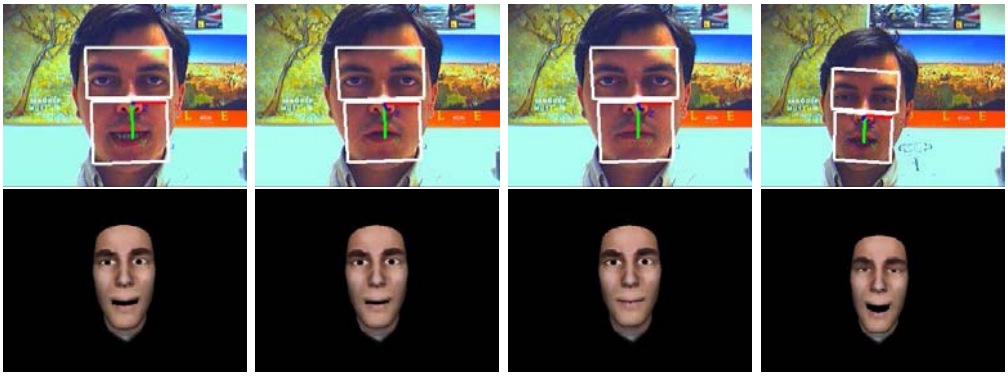


Figure 5: Human face re-animation (motion capture for animation) using appearance based techniques. In the upper row we show some images of the sequence. The tracked regions are shown in white. In the lower row we show the result of animating a graphical model with the parameters estimated by the tracker (see [16] for details).

We have proved that the previous factorisation can be made for rotation-translation-scale, affine, and projective motion models [9]. In Fig. 5 we show the results of tracking a face with this procedure, employing the estimated parameters to animate a graphical model of the head [16].

5 Shape+appearance models

In the appearance-based models presented in the previous section, the variation of the gray level of a pixel is caused by the combination of a nonrigid deformation of the object with changes in other sources of variability (e.g. illumination, pose, etc.). The combination of different sources of variation makes, in general, the appearance subspace be non-linear. To solve this problem the shape+appearance approach separates the variations due to nonrigid motion from other sources. In the case of the human face, two linear subspaces are used: one modelling the deformations of the shape of the face and another modelling the variations in appearance (texture). Both subspaces are linear and PCA models can be used, achieving good generalisation capabilities [23].

The shape+deformation approach has been used with 2D and 3D models of the face. The *Active Appearance Models* (AAMs) [10] are based on 2D models of the shape of the face combined with a 2D model of the texture (appearance). The *3D Morphable Models* [6] define a 3D model of face shape and a 2D model for face texture. In both models a shape is represented by the coordinates of a set of points on the object, $\mathbf{S} = (\mathbf{x}_1^\top, \mathbf{x}_2^\top, \ldots, \mathbf{x}_n^\top)^\top$. In the planar (2D) models each $\mathbf{x}_i \in \Re^2$ represents a position in the image plane, while in the 3D models, $\mathbf{x}_i \in \Re^3$, is a 3D point in space. A dense shape is usually obtained by interpolating between the three nearest neighbours (see Fig. 6). The appearance of the face is represented with an integer vector, $\mathbf{A}$, that represents the grey level (or colour) of each point of the face in the mean shape $\bar{\mathbf{S}}$. The shape, $\mathbf{S}_t$, and appearance, $\mathbf{A}_t$, of an object at time instant t will be generated from two linear models

$$\mathbf{S}_t = \bar{\mathbf{S}} + \mathtt{B}_s \mathbf{c}_s; \quad \mathbf{A}_t = \bar{\mathbf{A}} + \mathtt{B}_a \mathbf{c}_a.$$

where $\bar{\mathbf{A}}$ is the mean appearance, $\mathtt{B}_s$ and $\mathtt{B}_a$ are respectively the shape and appearance linear subspace basis. Finally $\mathbf{c}_s$ and $\mathbf{c}_a$ are respectively the shape and appearance model parameters.

The 3D motion of a point is the composition of a rigid motion caused by the translation and rotation of the object in space and a non-rigid motion caused by the deformation of the object. Any configuration of the object in 3D space can be generated with a motion model, f, which moves and deforms each of its 3D points,

$$\mathbf{x}'_i = f(\mathbf{x}_i, \boldsymbol{\mu}, \mathbf{c}_s) = \mathtt{R}(\mathbf{x}_i + [\mathtt{B}_s \mathbf{c}_s](\mathbf{x}_i)) + \mathbf{t}, \tag{9}$$

where $\boldsymbol{\mu} = (\alpha, \beta, \gamma, t_x, t_y, t_z)^\top$ is the vector of rigid motion parameters.

We have developed a human face tracking system based on a 3D shape+appearance model like the one introduced in equation (9) [20]. Our texture model is composed by a set of small planar patches and a basis $\mathtt{B}_a$ that represents the illumination variations of the patch grey levels (see Fig. 6). Each patch is tangent to the 3D volume of the target object and its texture is the result of projecting the texture of the object orthogonally from the surface of the object.

In this case, equation (1) takes the form

$$\arg \min_{\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a} E(\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a), \tag{10}$$

given

$$E(\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a) = ||\mathbf{I}(p(\mathbf{x}_i, \mathbf{q}), 0) - \mathbf{I}(p(f(\mathbf{x}_i, \boldsymbol{\mu}, \mathbf{c}_s), \mathbf{q}), t) + \mathtt{B}_a(\mathbf{c}_a - \delta\mathbf{c}_a)||^2, \tag{11}$$

where $\mathbf{x}_i \in \Re^3$, $\boldsymbol{\mu}$ is the rigid motion parameter vector, $f(\mathbf{x}, \boldsymbol{\mu}, \mathbf{c}_s)$ is the motion model of the face and $p(\mathbf{x}, \mathbf{q})$ is a projection function with projection parameters $\mathbf{q}$. Here we make no assumption as to which projection model is used, although in our experiments we assume a projective camera.

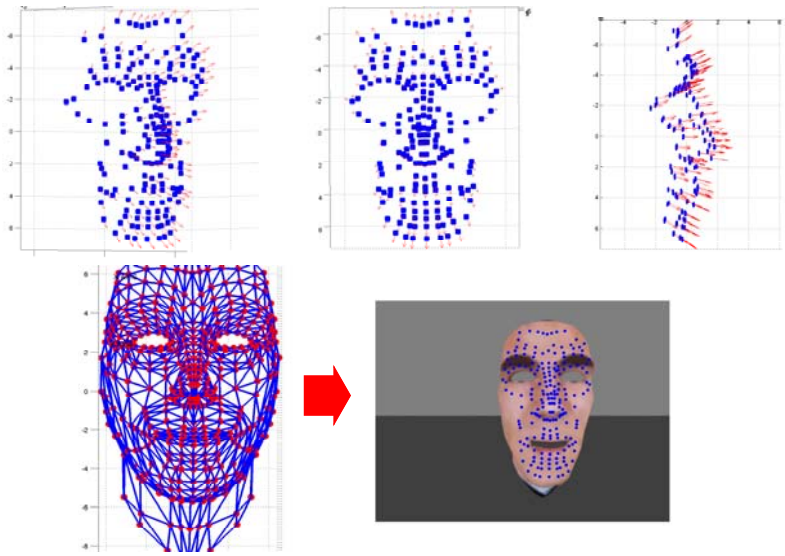


Figure 6: Patch based 3D face model. In the lower row are shown the projection of the model patches onto an image and a triangle mesh used to densely estimate the 3D position of the face vertices.

5.1 Minimising $E(\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a)$

Tracking amounts to find, for each time instant t, the set of parameters for which equation (11) is minimum. However, as it was stated in section 2, solving this problem has a high computational cost, which is a great inconvenience for real-time applications. In [1] an efficient solution is presented to solve this problem, the so-called *Inverse Compositional Algorithm*.

Equation (11) can be rewritten so that the increment to the parameters is referred to the reference texture, $\mathbf{I}(\mathbf{x}_i, 0)$

$$E(\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a) = ||\mathbf{I}(p(f(\mathbf{x}_i, \delta\boldsymbol{\mu}, \delta\mathbf{c}_s), \mathbf{q}), 0) - \mathbf{I}(p(f(\mathbf{x}_i, \boldsymbol{\mu}, \mathbf{c}_s), \mathbf{q}), t) + \mathtt{B}_a(\mathbf{c}_a - \delta\mathbf{c}_a)||^2. \tag{12}$$

Making a first order Taylor expansion in the equation above we have

$$E(\delta\boldsymbol{\mu}, \delta\mathbf{c}_s, \delta\mathbf{c}_a) = ||\mathbf{I}(p(\mathbf{x}_i, \mathbf{q}), 0) - \mathbf{I}(p(f(\mathbf{x}_i, \boldsymbol{\mu}, \mathbf{c}_s), \mathbf{q}), t) - \mathtt{M}_0\delta\mathbf{p}||^2, \tag{13}$$

where $\mathtt{M}_0$ is a matrix that represents the Jacobian of the reference texture with respect to the motion and appearance parameters. Note that $\mathtt{M}_0$ is constant (as it is evaluated at $\boldsymbol{\mu} = \mathbf{0}$, $\mathbf{c}_s = \mathbf{0}$ and $\mathbf{c}_a = \mathbf{0}$) and its pseudo-inverse can be precomputed off-line. This is the key for the efficiency of this algorithm.

Finally (13) can be solved linearly using least-squares:

$$\delta\mathbf{p} = \begin{bmatrix} \delta\boldsymbol{\mu} \\ \delta\mathbf{c}_s \\ \delta\mathbf{c}_a \end{bmatrix} = (\mathtt{M}_0^\top \mathtt{M}_0)^{-1}\mathtt{M}_0^\top \mathcal{E}(t+\delta t).$$

Model parameters are updated according to the following formulae [20]:

$$\begin{aligned} \mathtt{R}(t+\delta t) &= \mathtt{R}(t)\delta\mathtt{R}^\top, \\ \mathbf{t}(t+\delta t) &= \mathbf{t}(t) - \mathtt{R}(t)\delta\mathtt{R}^\top\delta\mathbf{t} \\ \mathbf{c}_s(t+\delta t) &= \mathbf{c}_s(t) - \delta\mathbf{c}_s \\ \mathbf{c}_a(t+\delta t) &= \mathbf{c}_a(t) - \delta\mathbf{c}_a, \end{aligned}$$

where $\mathtt{R}$ and $\mathbf{t}$ are, respectively, the rotation and translation of the target object with respect to a reference pose.

5.2 Results

In order to evaluate our algorithm empirically, we have set up experiments with a synthetic image sequence. We have developed a framework for creating synthetic sequences of a deforming head model. The head model is based on a previous work by Parke et. al. [21] which includes 512 vertices and encodes 18 different muscles of the face. Using a ray-tracer we simulate a projective camera located at 20 units of distance from the head model, which is has a depth of 5 units. To the left of the scene we have placed a light source, pointing directly towards the head. In Fig. 7 we show some key frames and associated tracking results from a 300 frames synthetic sequence. Results show that both motion and texture parameters are accurately estimated even when there are quite noticeable changes in illumination and facial expressions.

6 Conclusions

In this paper we have made an introduction to facial expression analysis techniques based on 2D appearance-based and 3D shape+texture approaches. At the same time, we have also presented some results from the PCR group of the UPM. Research in this field has considerably advanced in the past years. Although there are available various systems capable of tracking a face in real-time (see, e.g. [9, 19]), there are still many open research issues such as tracking initialisation, follow-up procedures after a complete loss of the target, and automating model construction, in the case of shape+texture approaches.

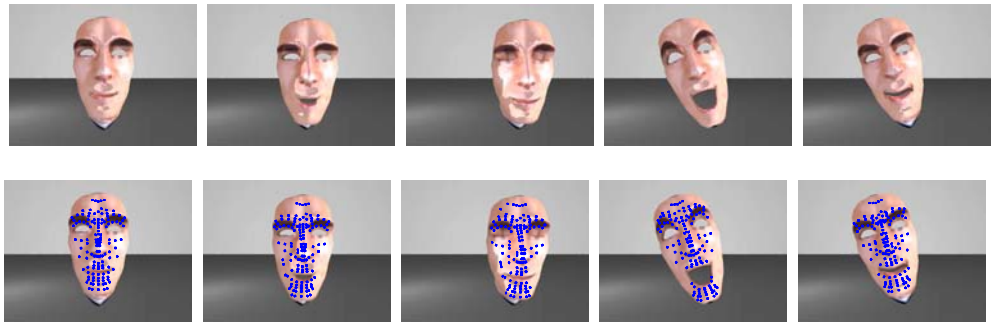


Figure 7: Tracking results with the 3D nonrigid tracker

The main advantage of 2D appearance-based techniques is the simplicity of the model. There are various procedures for automatically learning linear appearance models [11, 15]. Unfortunately, the manifold of facial appearance is non-linear. In consequence these simple linear models have quite low generalisation capabilities. There are also automated procedures for learning non-linear subspace models [18] and for probabilistically representing the dynamics of appearance variation [22, 12].

On the other hand, techniques based on shape+appearance models produce face representations which are more precise than those obtained using exclusively 2D appearance-based techniques. Since the shape and texture spaces are linear, these models generalise quite well. The main drawback of shape+texture approaches is that they have complex training procedures which often require manual intervention [7, 2].

Acknowledgements

This work has been funded by the spanish ministry of Science and Technology under project TIC2002-00591. Enrique Muñoz was funded by a FPU grant from the Spanish Ministry of Education.

References

[1] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. of CVPR*, volume 1, pages 1090–1097. IEEE, 2001.

[2] Simon Baker, Iain Matthews, and Jeff Schneider. Automatic construction of active appearance models as an image coding problem. *Trans. on PAMI*, 26(10):1380–1384, October 2004.

[3] B. Bascle and A. Blake. Separability of pose and expression in facial tracing and animation. In *Proc. of ICCV*, pages 323–328. IEEE, 1998.

[4] S. Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Proc. of ICPR*, 1996.

[5] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of SIGGRAPH*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[7] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Trans. on PAMI*, 25(9):1–12, September 2003.

[8] José M. Buenaposada and Luis Baumela. Real-time tracking and estimation of plane pose. In *Proc. of ICPR*, volume II, pages 697–700, Quebec, Canada, August 2002. IEEE.

[9] José M. Buenaposada, Enrique Muñoz, and Luis. Baumela. Efficient appearance-based tracking. In *Proc. CVPR-Workshop on Nonrigid and Articulated Motion*, volume 1. IEEE, June 2004.

[10] T. Cootes, G.J. Edwards, and C. Taylor. Active appearance models. In *Proc. of ECCV*. Springer-Verlag, 1998.

[11] Fernando de la Torre and Michael J. Black. Robust parameterized component analysis: Applications to 2d facial modeling. In *Proc. European Conference on Computer Vision (4)*, volume 2353 of *Lecture Notes on Computer Science*, pages 653–669. Springer, 2002.

[12] Huang Fei and Ian Reid. Joint bayes filter: A hybrid tracker for non-rigid hand motion recognition. In *Proc. of ECCV*, volume 3023 of *Lecture Notes on Computer Science*, pages 497–508, 2004.

[13] Andrew Gee and Roberto Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, 1996.

[14] Gregory Hager and Peter Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Trans. on PAMI*, 20(10):1025–1039, 1998.

[15] Lim Jongwoo, David Ross, Lin Ruei-Sung, and Yang Ming-Hsuan. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, 2004.

[16] Luis Baumela José Miguel Buenaposada, Enrique Muñoz. Performance driven facial animation by appearance based tracking. In *Proc. of Iberian Conference on Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes on Computer Science*, pages 476–483. Springer, 2005.

[17] Marco La Cascia, Stan Sclaroff, and Vasili Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *Trans. on PAMI*, 22(4), April 2000.

[18] Kuang-Chih Lee and David Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Proc. of CVPR*, 2005.

[19] Iain Matthews and Simon Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[20] Enrique Muñoz, José M. Buenaposada, and Luis Baumela. Efficient model-based 3d tracking of deformable objects. In *Proc. of ICCV*, volume I, pages 877–882, 2005.

[21] Frederick I. Parke and Keith Waters. *Computer Facial Animation*. AK Peters Ltd, 1996.

[22] Kentaro Toyama and Andrew Blake. Probabilistic tracking in a metric space. In *Proc. of ICCV*, 2001.

[23] Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. *Trans. on PAMI*, 19(7):733–742, 1997.

Empirical study of multi-scale filter banks for object categorization

Manuel J. Marín-Jiménez, Nicolás Pérez de la Blanca
University of Granada.
Department of Computer Science and Artificial Intelligence.
Periodista Daniel Saucedo Aranda, S/N. Granada, Spain

Abstract

The aim of this work is the evaluation of different multi-scale filter banks, mainly based on oriented Gaussian derivatives and Gabor functions, to be used in the generation of robust features for visual object categorization. In order to combine the responses obtained from several spatial scales, we use the biologically inspired HMAX model [1]. We have tested the different sets of features on the challenging Caltech 101-object categories database, and we have performed the categorizarion procedure with AdaBoost, Support Vector Machine and JointBoosting classifiers. Features based on second order Gaussian derivatives, combined with JointBoosting classifiers, achieve a 46.3% correct classification rate over the Caltech-101 database.

Keywords: object categorization, feature extraction, filter banks, Gaussian derivatives, Gabor.

1 Introduction

The Marr's theory [2] supports that in the early stages of the vision process, there are cells that respond to stimulus of primitive shapes, such as corners, edges, bars, etc. Young [3] models these cells by using Gaussian derivative functions. Riesenhuber & Poggio [1] propose a model for simulating the behavior of the Human Visual System (HVS), at the early stages of vision process. This model, named HMAX, generates features that exhibit interesting invariance properties (illumination, position, scale and rotation). More recently, Serre et al. [4], based on HMAX, propose a new model for image categorization adding to the HMAX model a learning step and changing the original Gaussian filter bank by a Gabor filter bank. They argue that the Gabor filter is much more suitable in order to detect local features. Nevertheless no experimental support has been given.

Different local feature based approaches are used in the field of object categorization in images. Serre et al. [4] use local features based on filter responses to describe

objects, achieving a high performance in the problem of object categorization. On the other hand, different approaches using grey-scale image patches, extracted from regions of interest, to represent parts of objects has been suggested, Fei-Fei et al. [5], Agarwal et al. [6], Leibe [7]. But, at the moment, there is not a clear advantage from any of these approaches. However, the non-parametric and simple approach followed by Serre *et al.* [4] in his learning step suggests that a lot of discriminative information can be learnt from the output of filter banks. Computing anisotropic Gabor features is a heavy task that only is justified if the experimental results show a clear advantage on any other type of filter bank.

The aim of this work is to carry out an experimental study in order to propose a new set of simpler filter banks, comparing the local features based on a Gabor filter banks with the ones based on Gaussian derivative filter banks. These features will be applied to the object categorization problem.

In section 2 of this paper, we review the use of Gaussian functions as local descriptors. In section 3, we introduce the proposed filter banks for object categorization. In section 4, we describe the experiments and present the experimental results. And finally, in section 5, we present the summary and our conclusions.

2 Using filters to describe images

Koenderink *et al.* [8] propose a methodology to analyze the local geometry of the images, based on the Gaussian function and its derivatives. Several optimization methods are available to perform efficient filtering with those functions [9, 10, 11]. Furthermore, *steerable* filters [12, 13] (oriented filters whose response can be computed as linear combination of other responses) can be defined based on Gaussian functions.

Yokono & Poggio [14] show, empirically, the excellent performance achieved by features created with filters based on Gaussian functions, applied to the problem of object recognition. In other published works, as Varma *et al.* [15], Gaussian filter banks are used to describe textures.

Our goal is to evaluate the capability of different filter banks, based on Gaussian functions, for encoding information usable for object categorization. We will use the biologically inspired HMAX model to combine responses of filters at different scales. In particular, HMAX consists of 4 types of features: S1, C1, S2 and C2. S1 features are the lowest level features, and they are computed as filter responses, grouped into scales; C1 features are obtained by combining pairs of S1 scales with the maximum operation; and, finally, C2 are the higher-level features, which are computed as the

maximum value of S2 from all the positions and scales. Where S2 features [1] measure how good is the matching of one C1 feature in a target image.

3 Our proposed multi-scale filter banks

Due to the existence of a large amount of works based on Gaussian filters, we propose to use, in the first level of the HMAX method, filter banks compound by the Gaussian function and its oriented derivatives.

The functions used in this work are defined by the following equations:

a) Isotropic Gaussian:

$$G^0(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{1}$$

b) First order Gaussian derivative:

$$G^1(x,y) = -\frac{y}{2\pi\sigma_x\sigma_y^3}\exp\left(-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}\right) \tag{2}$$

c) Second order Gaussian derivative:

$$G^2(x,y) = \frac{y^2-\sigma_y^2}{2\pi\sigma_x\sigma_y^5}\exp\left(-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}\right) \tag{3}$$

d) Laplacian of Gaussian:

$$LG(x,y) = \frac{(x^2+y^2-2\sigma^2)}{2\pi\sigma^6}\cdot\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4}$$

In order to improve the information provided by the features, we propose to include, in the lowest level, the responses of the Forstner operator [16], used to detect regions of interest. For each image point, we can compute a q value, in the range $[0,1]$, by using equation 6.

$$N(x,y) = \int_W M(x,y)dxdy \approx \Sigma M_{i,j} \tag{5}$$

$$q = 1 - \left(\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2}\right)^2 = \frac{4detN}{(trN)^2} \tag{6}$$

[1] Let P_i and X be patches, of identical dimensions, extracted at C1 level from different images, then, $S2$ is defined as: $S2(P_i,X) = \exp(-\gamma\cdot\|X-P_i\|^2)$, where γ is a tunable parameter.

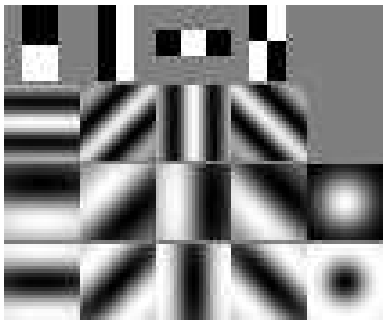


Figure 1: Sample filter banks. From top to bottom: Viola, Gabor, first derivative of Gaussian with a zero-order Gaussian, and second derivative of Gaussian with a Laplacian of Gaussian. Orientations: 0, 45, 90 and 135

Where M is the moments matrix and W is the neighborhood of the considered point (x, y).

We will compare our proposed filter banks with the filter banks based on Gabor functions (as defined in [4]). On the other hand, Viola and Jones, in their fast object detector [17] use filters, which are simplified versions of first and second order Gaussian derivative filters, to extract local features. Since those filters achieve very good results and are computable in a very efficient way, we will include them in our comparison.

Figure 1 shows sample filter banks: Viola, Gabor, first order Gaussian derivatives with an isotropic zero-order Gaussian, and second order Gaussian derivatives with an isotropic Laplacian of Gaussian.

4 Experiments

We have chosen the Caltech 101-object categories [2] to perform the experiments. This database has become, nearly, the standard database for object categorization. It contains images of objects grouped into 101 categories, plus a background category commonly used as the negative set. This is a very challenging database because the objects are embedded in cluttered backgrounds and have different scales and poses. Figure 2 shows some sample images drawn from diverse categories of this database. In order to make a robust comparison, we have discarded the 15 categories that contains less than 40 samples. All the images were normalized in size, so that the

[2]The Caltech-101 database is available at *http://www.vision.caltech.edu/*

Figure 2: Samples from diverse categories of the Caltech-101 database.

longer side had 140 pixels and the other side was proportional, to preserve the aspect ratio.

4.1 Multi-scale filter banks evaluation

We will compute biologically inspired features based on different filter banks. For each feature set, we will train binary classifiers for testing the presence or absence of objects in images from a particular category. The set of the negative samples is compound by images of all categories but the current one, plus images from the background category. This strategy differs from the classic one, where the negative set is compound only by background images, because we are interested in studying the capability of the features to distinguish between different categories, and not only in distinguishing foreground from background.

The eight filter banks defined for this experiment are the following:

(1)	Viola (2 edge filters, 1 bar filter and 1 special diagonal filter);
(2)	Gabor (as [4]);
(3)	anisotropic first-order Gaussian derivative;
(4)	anisotropic second-order Gaussian derivative;
(5)	(3) with an isotropic zero-order Gaussian;
(6)	(3) with a Laplacian of Gaussian and Forstner operator;
(7)	(3), (4) with a zero order Gaussian, Laplacian of Gaussian and Forstner op;
(8)	(4) with Forstner operator.

The Gabor filter and the anisotropic first and second order Gaussian derivatives (with aspect-ratio equals 0.25) are oriented at 0, 45, 90 and 135. All the filter banks contain 16 scales (as [4]).

The standard deviation used for the Gaussian-based filter banks is equal to a quarter of the filter-mask size. Table 1 shows the value of the parameters for the filter banks, where FS is the size (in pixels) of the 16 mask-filters and σ is the related standard deviation of the functions.

FS	7	9	11	13	15	17	19	21
σ	1.75	2.25	2.75	3.25	3.75	4.25	4.75	5.25
FS	23	25	27	29	31	33	35	37
σ	5.75	6.25	6.75	7.25	7.75	8.25	8.75	9.25

Table 1: Filter mask size (FS) and filter width (σ) for Gaussian-based filter banks.

In these filter banks we have combined linear filters (Gaussian derivatives of different orders) and non-linear filters (Forstner operator), in order to study if the mixture of information of diverse nature enhances the quality of the features.

We will generate features (named C2) following the HMAX method and using the same empirical tuned parameters proposed by Serre *et al.* in [4]. The evaluation of the filters will be done following a strategy similar to the one used in [5]. From one single category, we draw 30 random samples for training, and 50 different samples for test, or less (the remaining ones) if there are not enough in the set. The training and test negative set are both compound by 50 samples, randomly chosen following the strategy previously explained. For each category and for each filter bank we will repeat 10 times the experiment.

Results During the patch [3] extraction process, we have always taken the patches from a set of prefixed positions in the images. Thereby, the comparison is straightforward for all filter banks. We have decided, empirically (fig. 3), to use 300 patches (features) per category and filter bank. If those 300 patches were selected (from a huge pool) for each individual case, the individual performances would be better, but the comparison would be unfair.

In order to avoid a possible dependence between the features and the type of classifier used, we have trained and tested, for each repetition, two different classifiers: AdaBoost (with decision stumps) [18] and Support Vector Machine (linear)

[3] In this context, a *patch* is a piece of a filtered image, extracted from a particular scale. It is three dimensional: for each point of the patch, it contains the responses of all the different filters, for a single scale.

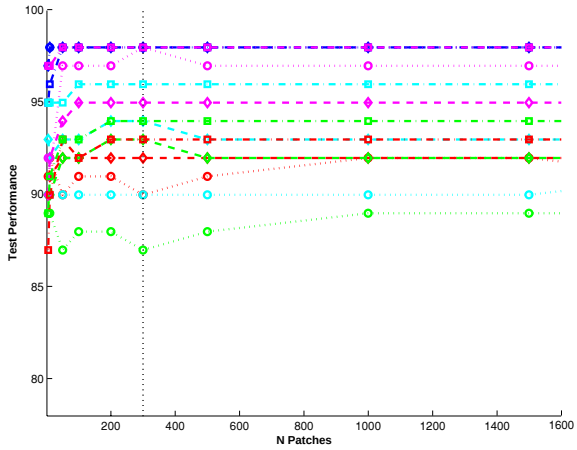


Figure 3: Evolution of performance versus number of patches. Evaluated on five sample categories (*faces, motorbikes, car-side, watch, leopards*), by using three different filter banks: Gabor, first order Gaussian derivative and second order Gaussian derivative. About 300 patches, the achieved performance is nearly steady.

[19].

For training the AdaBoost classifiers, we have set two stop conditions: a maximum of 300 iterations (as many as features), or a training error rate lower than 10^{-6}. On the other hand, for training the SVM classifiers, we have selected the parameters through a cross-validation procedure.

The results obtained for each filter bank, from the classification process, are summarized in table 2. For each filter bank, we have computed the average of the all classification ratios, achieved for all the picked out categories, and the average of the confidence intervals (of the means). The top row refers to AdaBoost and the botton row refers to Support Vector Machine. The performance is measured at *equilibrium-point* (when the miss-ratio equals the false positive ratio).

-	*Viola*	*Gabor*	*FB-3*	*FB-4*	*FB-5*	*FB-6*	*FB-7*	*FB-8*
AdaB	78.4 , 4.3	**81.4** , 3.9	81.2, 3.9	**81.4** , 4.2	**81.9** , 3.3	77.9 , 4.5	80.3 , 4.3	78.1, 4.0
SVM	84.2 , 2.3	**85.5** , 2.5	84.1 , 3.6	**86.0** , 3.3	84.1 , 3.0	82.6 , 2.7	82.8 , 2.4	82.7, 2.6

Table 2: Results of classification using different filter banks: averaged performance and averaged confidence intervals. First row: AdaBoost. Second row: SVM linear.

Figure 4 shows the averaged performance achieved, for the different filter banks,

by using AdaBoost and SVM. In general, by using this kind of features, SVM outperforms AdaBoost.

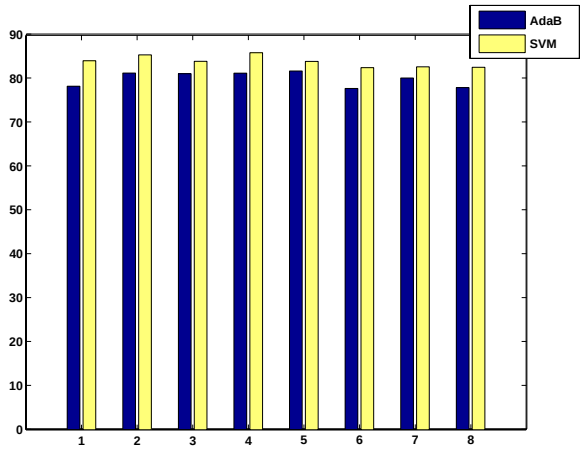


Figure 4: Comparing the filter banks with AdaBoost and SVM classifiers. From left to right: (1) Viola, (2) Gabor, (3) 1st deriv., (4) 2nd deriv, (5) 1st deriv. with 0 order, (6) 1st deriv. with LoG and Forstner op., (7) G0, 1oGD, 2oGD, LoG, Forstner, (8) 2oGD and Forstner.

If we focus on table 2, we see that the averaged performances are very similar. Also, the averaged confidence intervals are overlapped. If we pay attention only at the averaged performance, the filter bank based on second order Gaussian derivatives, stands out slightly from the others.

So, our conclusion for this experiment is that Gaussian filter banks represent a clear advantage in comparison to the Gabor filter bank. It is much better in terms of computational burden and is slightly better in terms of categorization efficacy. However, depending on the target category, one filter bank may be more suitable than other.

4.2 Multicategorization

In this experiment, we deal with the problem of multicategorization on the full Caltech 101-object categories, included the background category. The training set is compound by the mixture of 30 random samples drawn from each category, and the test set is compound by the mixture of 50 different samples drawn from each category (or the remaining, if it is less than 50). Each sample is enconded by using 4075 patches [4], randomly extracted from the full training set. These features are

computed by using the oriented second order Gaussian derivative filter bank.

In order to perform the categorization process, we will use a Joint Boosting classifier, proposed by Torralba *et al.* [20]. Joint Boosting trains, simultaneously, several binary classfiers which share features between them, improving this way the global performance of the classification.

Under these conditions, we have achieved an average 46.3% of global correct categorization (chance is below 1% for this database), where more than 40 categories are over 50% of correct categorization. By using only 2500 features, the performance is about 44% (fig. 5.c). On the other hand, if we use 15 samples per category for training, we achieve a 39.5% rate. Figure 5 shows the confusion matrix for the 101 categories plus background (by using 4075 features and 30 samples per category). For each row, the highest value should belong to the diagonal.

Other results on this database, using diverse technics, are: Serre 42% [4], Holub 40.1% [21], Grauman 43% [22], and, the best result up to our knowledge, Berg 48% [23].

Figure 5.b shows the histogram of the individual performances achieved for the 101 object categories, in the multiclass task. Note, that only 6 categories shows a performance lower than 10%, and 17 categories are over 70%.

In figure 5.c, we can see the evolution of the test performance, depending on the number of patches used for encode the samples. With only 500 patches, the performance is about 31%. If we use 2500 patches, the performance increases up to 44%.

Figure 5.d shows how the training error evolves, yielded by the Joint-Boosting classifier, over the 101-object categories. The error decreases with the number of iterations following a logarithmic behavior.

Figure 6.a shows how the first 50 features selected by JointBoosting, for the joint categorization of the 101 categories, are shared between the 102 categories (background is included as a category). The rows represent the features and the columns are the categories. A black-filled cell means that the feature is used to represent the category.

Figure 6.b shows the first four features selected by JointBoosting, for the joint categorization of the 101 object categories. The size of the first patch is 4x4 (with 4 orientations), and the size of the others is 8x8 (with 4 orientations).

In table 3, we show which categories share the first 10 selected patches. Three of the features are used only by one single category.

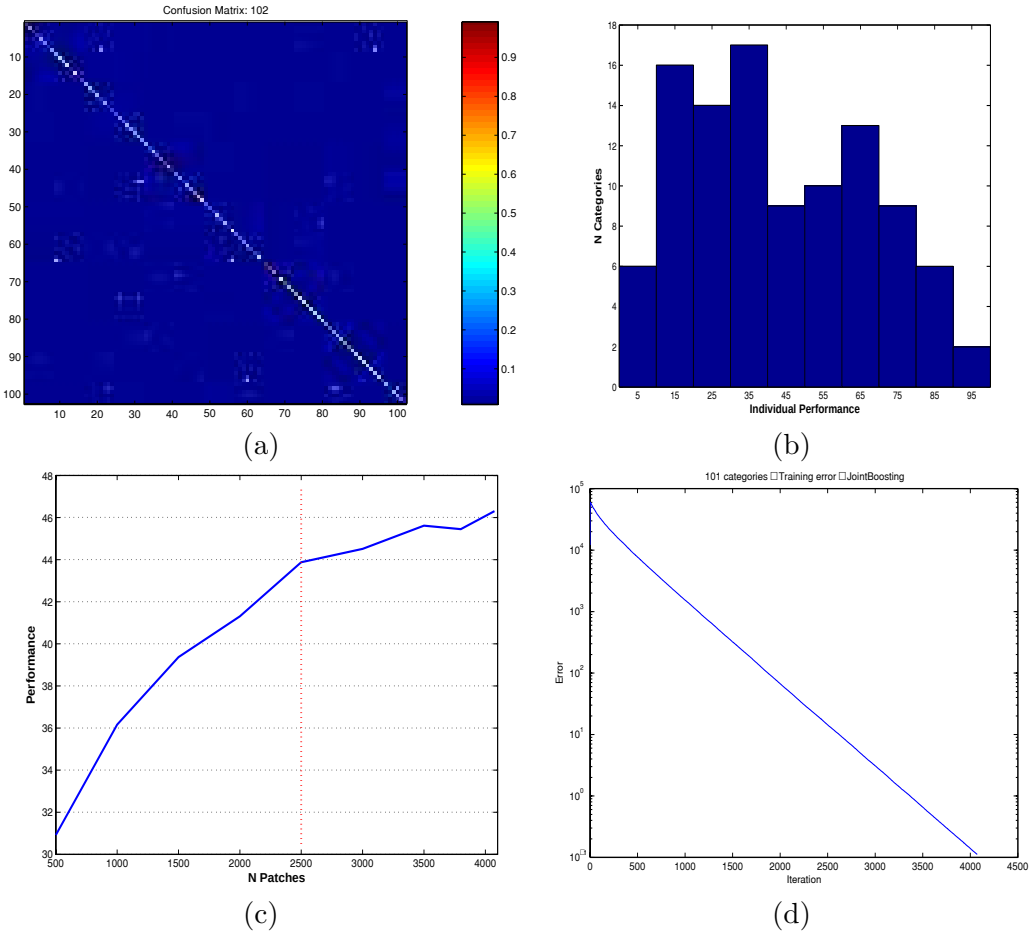


Figure 5: 101 object categories learnt with 30 samples per category and JointBoosting classifier. (a) Confusion matrix for 101-objects plus background class. Global performance is over 46%. (b) Histogram of individual performances. (c) Global test performance vs Number of features. (d) Training error yielded by Joint Boosting. Y-axis: logarithmic.

# Feature	Shared-Categories
1	yin yang
2	car side
3	pagoda, accordion
4	airplanes , wrench , ferry , car side , stapler , euphonium , mayfly , scissors , dollar bill , mandolin , ceiling fan , crocodile , dolphin
5	dollar bill, airplanes
6	trilobite , pagoda , minaret , cellphone , accordion
7	metronome , schooner , ketch , chandelier , scissors , binocular , dragonfly , lamp
8	Faces easy
9	inline skate , laptop , buddha , grand piano , schooner , panda , octopus , bonsai , snoopy , pyramid , brontosaurus , background , gramophone , metronome
10	scissors , headphone , accordion , yin yang , saxophone , windsor chair , stop sign , flamingo head , brontosaurus , dalmatian , butterfly , chandelier , binocular , cellphone , octopus , dragonfly , Faces , wrench

Table 3: First 10 shared features by categories.

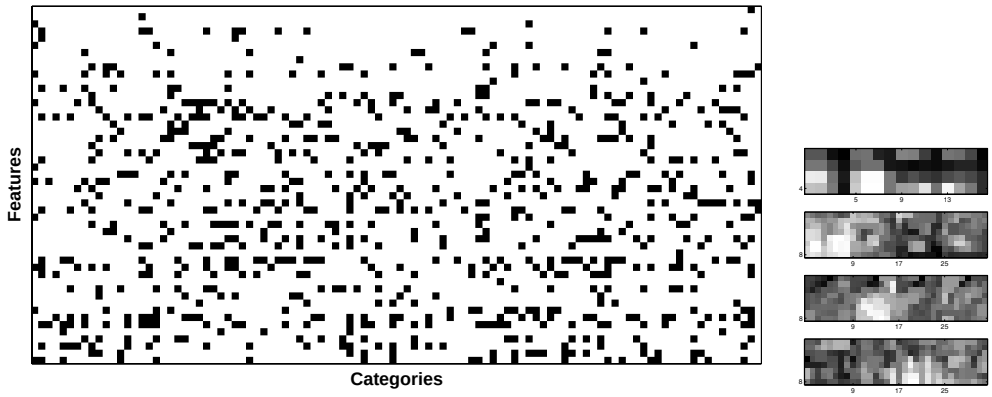
4.2.1 Caltech selected categories database.

In this section, we focus on a subset of the Caltech categories: *motorbikes, faces, airplanes, leopards* and *car-side*.

The filter bank used for these experiments is based on second order Gaussian derivatives, and its parameters are the same ones than in the previous sections. 2000 patches have been used to encode the samples.

Experiment 1 We have trained JointBoosting classifiers with an increasing number of samples (drawn at random), and tested with all the remaining ones. Figure 7 shows how the mean test performance, for 10 repetitions, evolves according to the number of samples (per category) used for training. On the left, we show the performance achieved when 4 categories are involved, and, on the right, when 5 categories are involved. With only 50 samples, these results are already comparable to the ones shown in [21].

Experiment 2 By using 4-fold cross-validation (3 parts for training and 1 for test), we have evaluated the performance of the JointBoosting classifier applied to the Caltech selected categories. The experiment is carried out with the 4 categories used in [24, 21] (all but *car-side*), and, also, with the five selected categories. Table 4 and table 5 contains, respectively, the confusion matrix for the categorization of the four and five categories. In both cases, individual performances (values of the diagonal) are greater than 97%, and the greater confusion-error is found when *airplanes* are classified as *motorbikes*. It is curious that the individual performances are slightly better for the 5-categories case.



(a) (b)

Figure 6: 101 object categories. (a) Left: first 50 shared features selected by JointBoosting. (b) Right: the first 4 features, selected by JointBoosting.

-	Motorbikes	Faces	airplanes	Leopards
Motorbikes	**99.75**	0.13	0.13	0
Faces	1.38	**98.62**	0	0
Airplanes	2.38	0	**97.50**	0.13
Leopards	0.50	0.50	0	**99.00**

Table 4: Caltech selected (as [24]). Mean performance from 4-fold cross-validation.

4.3 Towards the universal visual codebook

The goal of this experiment is to evaluate the capability of generalization of the features generated with HMAX and the proposed filter banks. In particular, we wonder if we could learn a category, without using patches extracted from samples belonging to it. For this experiment we will use the Caltech-7 database (*faces, motorbikes, airplanes, leopards, cars rear, leaves* and *cars side*), used in other papers [24]. Each category is randomly split into two separated sets of equal size, the training and test sets. For each instance of this experiment, we extract patches from all the categories but one, and we focus our attention on what happens with that category.

We have extracted 285 patches from each category, therefore each sample is encoded with 1710 (285×6) patches. We train a Joint Boosting classifier with the features extracted from 6 categories and test over the 7 categories. We repeat the procedure 10 times for each excluded category. The filter bank used for this

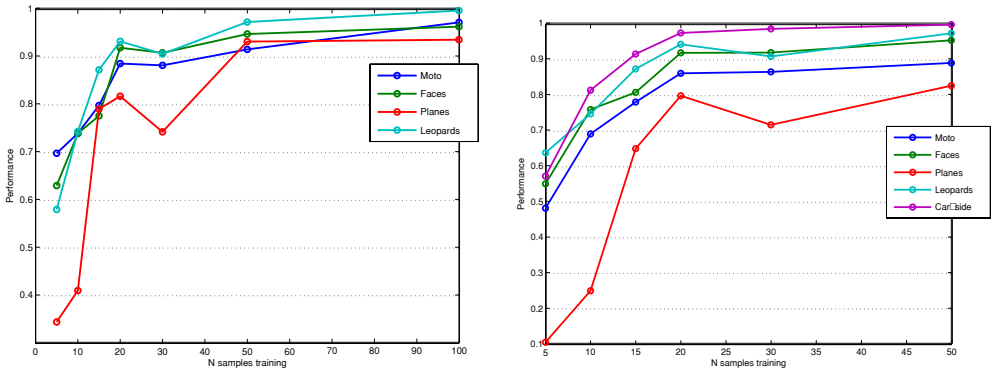


Figure 7: Performance versus number of training samples, in multicategorization environment. Left: 4 categories. Right: 5 categories.

-	Motorbikes	Faces	airplanes	Leopards	Car side
Motorbikes	**99.87**	0.13	0	0	0
Faces	1.15	**98.85**	0	0	0
Airplanes	2.00	0	**98.00**	0	0
Leopards	0.50	0.50	0	**99.00**	0
Car side	0.81	0	0	0.81	**98.37**

Table 5: Caltech selected (5 categories). Mean performance from 4-fold cross-validation.

experiment is compound by 4 oriented first order Gaussian derivatives, plus an isotropic Laplacian of Gaussian.

Table 6 shows the mean global multicategorization performance, and the individual performance, achieved for each excluded category. We can see that all the global results are near the 95% of correct categorization. These results suggest that there are features that are shared between categories in a 'natural' way, and hence it encourages the search for the universal visual codebook, proposed in some works [4].

5 Summary and discussion

An experimental study has been carried out in order to compare the performance of different filter banks for the object categorization problem. We have generated multi-scale features with eight proposed filter banks, which have been used to learn the object categories included in the challenging Caltech-101 database. The re-

	No-face	*No-moto*	*No-airp*	*No-leop*	*No-car_rear*	*No-leav*	*No-car_side*
Global	94.7	93.7	94.8	96.8	95.9	95	93.5
Individual	98.7	96.9	96.5	94.0	88.9	91.4	88.5

Table 6: Categorization by using non-specific features. First row shows the mean global performance (all categories) and, the second row shows the individual performance (just the excluded category). It seems that the *car rear* and *car side* categories need their own features to represent them in a better way.

sults show that the local features generated with filter banks based on Gaussian derivatives, achieve an excellent performance in the object categorization problem compared to the Gabor-based features. In fact, the results provided in the task of multicategorization on the Caltech-101, combined with JointBoosting classifiers, are very competitive compared to the state-of-the-art. However, we think that it is necessary to study alternative options, other than including more filter banks, to improve the achieved performance. On the other hand, we have noticed that Support Vector Machine classifiers, on average, works better than AdaBoost with this kind of features.

As a result, we can say that the trend of building large pools of visual features, to be shared between different object categories, seems a promising way to deal with the problem of general object categorization.

6 Acknowledgments

Thanks to Dr. Jordi Vitrià for his helpful comments. This work was partially supported by the Spanish Ministry of Education and Science (beca AP2003-2405).

References

[1] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[2] David Marr. *Vision*. W. H. Freeman and Co., 1982.

[3] Richard A. Young. The gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision*, 2(4):273–293, 1987.

[4] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *IEEE CSC on CVPR*, June 2005.

[5] F.F. Li, R. Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. 2004.

[6] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.

[7] Bastian Leibe. *Interleaved Object Categorization and Segmentation.* PhD thesis, ETH Zurich, October 2004.

[8] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[9] I. T. Young and L. J. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44(2):139–151, 1995.

[10] L. van Vliet, I. Young, and P. Verbeek. Recursive gaussian derivative filters. In *l4th International Conference on Pattern Recognition (ICPR-98)*, volume 1, pages 509–514. IEEE Computer Society Press, August 1998.

[11] Jan-Mark Geusebroek, Arnold W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. In *ECCV (1)*, pages 99–112, 2002.

[12] W.T. Freeman and E.H. Adelson. Steerable filters for early vision, image analysis and wavelet decomposition. In IEEE Computer Society Press, editor, *3rd Int. Conf. on Computer Vision*, pages 406–415, Dec 1990.

[13] P. Perona. Deformable kernels for early vision. *IEEE PAMI*, 17(5):488–499, May 1995.

[14] J. J. Yokono and T. Poggio. Oriented filters for object recognition: an empirical study. In *Proc. of the Sixth IEEE FGR*, May 2004.

[15] M. Varma and A. Zisserman. Unifying statistical texture classification frameworks. *Image and Vision Computing*, 22(14):1175–1183, 2005.

[16] W. Forstner and E. Gulch. A fast operador for detection and precise location of distinct points, corners and centres of circular features. *ISPRS Intercommission Workshop*, June 1987.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, volume 1, pages 511–518, 2001.

[18] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics. Stanford University, 1998.

[19] E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: training and applications. Technical Report AI-Memo 1602, MIT, March 1997.

[20] Antonio B. Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR (2)*, pages 762–769, 2004.

[21] Alex D. Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *ICCV05*, 2005.

[22] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the IEEE ICCV*, October 2005.

[23] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005.

[24] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *IEEE CVPR'03*, 2:264–271, Feb 2003.

Hierarchical-based Clustering using Local Density Information for Overlapping Distributions *

Damaris Pascual†, Filiberto Pla‡, J. Salvador Sánchez‡

† Dept de Ciencia de la Computación, Universidad de Oriente,
Av. Patricio Lumunba s/n, Santiago de Cuba, CP 90100, Cuba
dpascual@csd.uo.edu.cu
‡ Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I,
12071 Castelló, Spain,
{pla, sanchez}@lsi.uji.es

Abstract

Clustering techniques are widely used in many application fields like image analysis, data mining, and knowledge discovery, among others. In this work, we present a new clustering algorithm to find clusters of different sizes, shapes and densities, able to deal with overlapping cluster distributions and background noise. The algorithm is divided in two stages, in a first step; local density is estimated at each data point. This local density is used to initialize the clustering grouping the objects around the object of local maximum density (core point). In a second stage, a hierarchical approach is used by merging clusters according to the introduced cluster distance, also based on local density in-formation. Experimental results on synthetic and real databases show the validity of the proposed method.

Keywords: density based clustering, overlapped distributions.

1 Introduction

Clustering algorithms are techniques widely used to discover relevant distributions and relationships in databases. The problem of clustering can be defined as: Given n points belonging to a d-dimensional space, provided some measurement of similarity or dissimilarity, the aim is to divide these points into a set of clusters so that the simi-larity between

* This work has been partially supported by projects IST-2001-37306 from EU, TIC2003-08496 from the Spanish CICYT, and GV04A/705 from Generalitat Valenciana.

patterns belonging to the same cluster is maximized whereas the similarity between patterns of different clusters is minimized.

Basically, there are two approaches in clustering techniques: the partitional approach and the hierarchical approach [7]. The partitioning methods build a partition from the database of n objects in k clusters. These algorithms assume a priori knowledge about the number of classes in which the database must be divided. The K-means is the best known partitional algorithm.

Hierarchical methods consist of a sequence of nested data partitions in a hierarchical structure, which can be represented as a dendogram. There exist two hierarchical approaches: agglomerative and divisive. The first one can be described in the follow-ing way: initially each point of the database form a single cluster, and in each level, the two most similar clusters are joined, until either a single cluster is reached with all the data points, or some stopping condition is defined, for instance, when the distance between the clusters is smaller than certain threshold. In the divisive approach, the process is the other way around.

The Single Link (SL) and the Complete Link (CL) methods are the most well known hierarchical strategies [3]. Some hierarchical algorithms are based on proto-types selection, as CURE [4]. On the other hand, in density–based algorithms, the clusters are defined as dense regions, where clusters are separated by low density areas [5]. Some of the most representative ones of this approach are DBSCAN [1], KNNCLUST [7] and SSN [2] algorithms.

Some of the problems these algorithms fail to tackle are the fact that clusters are not completely separable, due to the overlapping of cluster distributions in usual real situations, and the presence of noisy samples. In this work we present an algorithm based on a hybrid strategy between the hierarchical and density-based approaches, with the aim of dealing with overlapped clusters and noisy samples, in order to discover the most signifcant density based distributions in the database.

2 Hierarchical Clustering using Local Probability Density

The objective of the algorithm here presented is to detect clusters of different shapes, sizes and densities even in the presence of noise and overlapping cluster distributions. The algorithm is a mixture of a density-based and a hierarchical-based approach, and it is divided in two stages. In the first stage, the initial clusters are constructed using a density-based approach. In a second stage, a hierarchical approach is used, based on a cluster similarity function defined in terms of cluster density measures and distances, joining clusters until either arriving to a pre-defined number or reaching a given stop-ping criterion.

2.1 Estimating Local Density

Let X be a set of patterns provided with a similarity measure between patterns d. Let x be an arbitrary element in the dataset and $R>0$. The neighbourhood VR of radius R of x is defined as the set:

$$V_R(x) = \{y / d(x,y) \leq R\}$$

and the local density p(x) of the non-normalized probability distribution at point x as:

$$p(x) = \sum_{i=1}^{N_x} \exp\left(- \frac{d^2(x, x_i)}{R^2} \right) \tag{1}$$

where x_i are the points that belong to the neighbourhood of radius R of x, VR.

In the algorithm we are going to differentiate between two concepts: core cluster and cluster. We will call core clusters to the sets that are obtained after applying the first stage of the algorithm, and we will call cluster to the sets of core clusters that will be grouped into clusters in a further stage.

2.2 Defining Cluster Similarities

As the objective of the second stage is to perform a hierarchical algorithm between the classes obtained in the first stage we need to define the di-similarity between two clusters.

Given two core clusters Ci and Cj , let us define the distance between them as:

$$d'(C_i, C_j) = \min \{d(x_i,x_j)\};\ \forall\ x_i, x_j / x_i \in C_i \text{ and } x_j \in C_j$$

Given two clusters Ki and Kj , let us define distance between them as:

$$\bar{d}(K_i, K_j) = \frac{P_c - P_m}{P_c}\ (1 + d''(K_i, K_j)) \tag{2}$$

where

$$d''(K_i, K_j) = \min \{d'(C', C'')\},\ \forall\ C', C'' / C' \in K_i \text{ and } C'' \in K_j$$

Given a core cluster C , let us define the centre of the core cluster to the point whose density is maximal within the core cluster. Let x' and x'' be the centres of C_i and C_j respectively. Therefore, let us define Pc as the minimum density of the core cluster centres x' and x'', that is,

$$P_c = \min(p(x'), p(x''))$$

In equation (2), P_m is the density of the midpoint between the two core clusters, that is, it is the midpoint of the border between both core clusters, which is defined as the midpoint between the nearest points xb_i and xb_j, one from each core cluster. To estimate the density of such a midpoint, it is interpolated from the density values of the mentioned points belonging to each cluster. To calculate the interpolated value, two different cases are taken into account when comparing the two neighbouring core clusters:

1. If $d'(C_i, C_j)>R$, the midpoint do not have points in its neighbourhood of radius R, then we take $P_m=0$ and the distance between the clusters becomes:

$$\bar{d}(K_i, K_j) = 1 + d''(K_i, K_j) \tag{3}$$

   That is, in the cases that clusters are well separated, the di-similarity measurement is given by the distance between the nearest core clusters.

2. If $d'(C_i, C_j)\leq R$, the midpoint has got points in its neighbourhood from both core clusters. In this case, the midpoint x is defined as either the border point xb_i from core cluster C_i or the border point xb_j from core cluster C_j. In order to avoid negative values in expression (2), the midpoint is chosen as the border point such as,

   if $P_c=p(x')$, then $x=xb_i$, and $P_m=p(xb_i)$

   else $x=xb_j$, and $P_m=p(xb_j)$

2.2 Grouping Core Clusters into Clusters

The clustering algorithm here presented consists of a hierarchical agglomerative strategy based on a Single Link approach, using the di-similarity measures defined in the previous Section. The use of such dissimilarity measures defines the behaviour of the clustering process and the response to the different local distributions of the patterns in the data set.

In a few words, the dissimilarity measure defined in (2) is aimed at considering that clusters are more similar when they probability distributions are either nearer in the feature space by means of a Single Link concept, or when their probability distributions are more overlapped. In the last case, when probability distributions are over-lapped ($d'=0$), the measure of similarity becomes the probability density term that appear in equation (2), which is a local estimate of the mixed probability distributions at the clusters border.

Therefore, the proposed algorithm can be summarized in two stages as follows:

First stage:

Input: radius R, data points and density noise threshold

Output: N core clusters

1. Initially, each point of the database is assigned to a single core cluster.
2. For each point x, calculate its neighbourhood of radius R, $VR(x)$
3. For each point x in the database, estimate its probability density $p(x)$ according to expression (1).
4. Assign each point x to the core cluster of the point x_c in its neighbourhood, being x_c the point with maximal density in the neighbourhood.
5. Mark all core clusters with density less than the density noise threshold as noise core clusters. The rest are the resulting N core clusters.

Second stage

Input: N core clusters

Output: K clusters

1. Initially, assign each core cluster from the first stage to a cluster. Therefore, there are N clusters with one core cluster.
2. Repeat until obtaining K clusters,
   2.1 Calculate the distance between each pair of clusters using expression (2)
   2.2 Join the clusters that their distance is minimum
3 Assign the noise core clusters to a nearest.

3 Experimental Results

In this section, some experimental results are presented aimed at evaluating the proposed algorithm, hereafter named H-density, and to compare it with some other similar algorithms referred in the introduction, DBSCAN, CURE and K-means. In order to test the algorithm, three groups of experiments are performed. The first one uses synthetic databases based on overlapped Gaussian distributions, in order to see the response of the proposed algorithm in these conditions. The second experiment uses two synthetic databases from [6], for comparison purposes, and to test the problem of the presence of noise, over-

lapping, and clusters of different sizes and shapes. Finally, some experiments are performed on two real databases.

3.1 Gaussian Databases

Four databases using Gaussian distribution were generated with different number of classes (Gaussian distributions), sizes and overlapping degrees. The number of samples and classes in each database is shown in Table 1.

Table 1. Gaussian databases generated used in the experiments.

Database	No samples	No classes
G1	4000	4
G2	6000	6
G3	6000	3
G4	8000	4

The results obtained with the proposed algorithm are shown in Figure 1, where we can notice how the algorithm has been able to correctly detect each one of the existing classes, even in the presence of significant overlapping (Figure 1 right).

The DBSCAN algorithm does not correctly detect all the classes in different databases because it is not able to separate the overlapped classes. For example, in data-base G2 it detects 3 classes for radius 5 and some noise points (Figure 2 left). If the radius is increased, it obtains three or less classes. If the radius decreases, the objects of the edge of the three classes are separated because they stay as noise points. This can be noticed in Figure 2 right, where the number of noise points increases. The others databases have a similar behaviour.

The CURE algorithm detects all the classes in databases G1, G3 and G4, but in database G2 it correctly detects three classes. However, in the case of six classes, it cannot detect the 4 classes that are highly overlapped. The same happens with the K-means algorithm, it detects the classes in databases G1, G3 and G4, but in the case of the database G2 the results depend on the initial centres (see Figure 3).

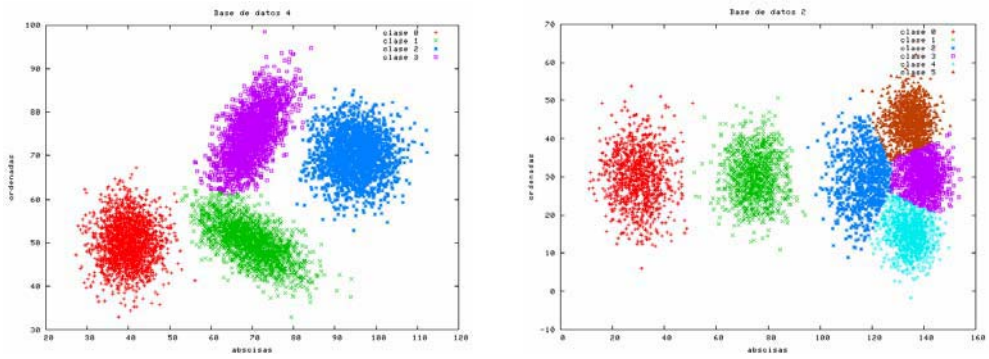


Figure 1. Results of the H-density algorithm on G4 (left) and G2 (right) databases.

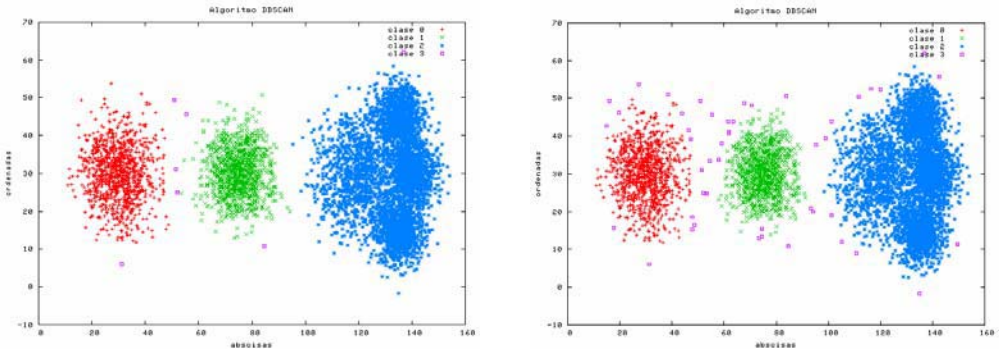


Figure 2. Results of the DBSCAN algorithm on G2 using radius=5, MinPts=4 (left), and radius=3, MinPts=4 (right).

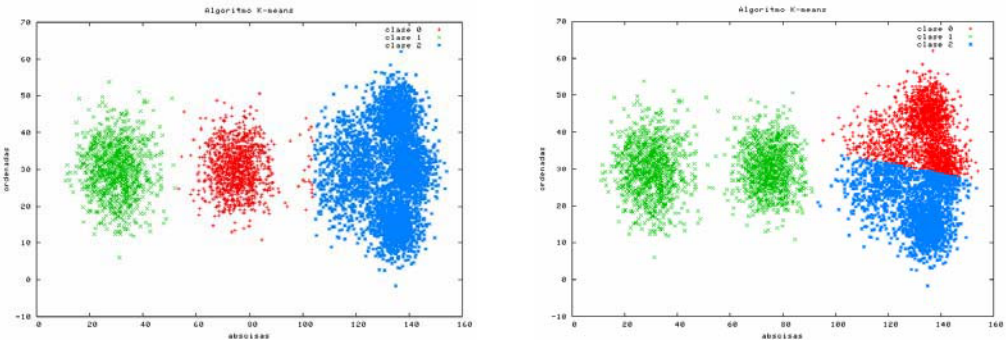


Figure 3. Results of the K-means algorithm on G2 with two different initializations.

3.2 Synthetic Databases

In [6], some experiments were presented for the DBSCAN and CURE algorithms using the databases of Figure 4 (see [6] for comparison results with those algorithms). Notice the presence of clusters of different shape, size, noise and overlapping. Figure 4 shows the result of applying the proposed H-density algorithm on these databases. Note how the algorithm has correctly grouped the main clusters present in the data set. Figure 5 shows the result of the K-means algorithm for 6 clusters (left) and 9 clusters (right) of the corresponding databases. The errors in the grouping are noticeable.

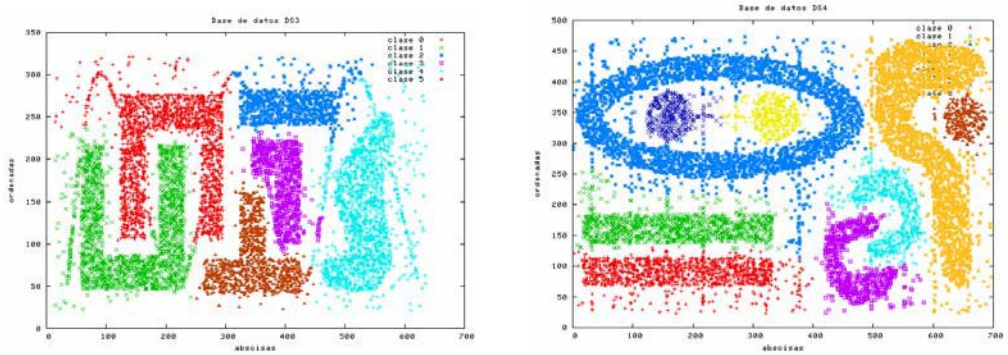


Figure 4. Results of the H-density algorithm on databases from [6].

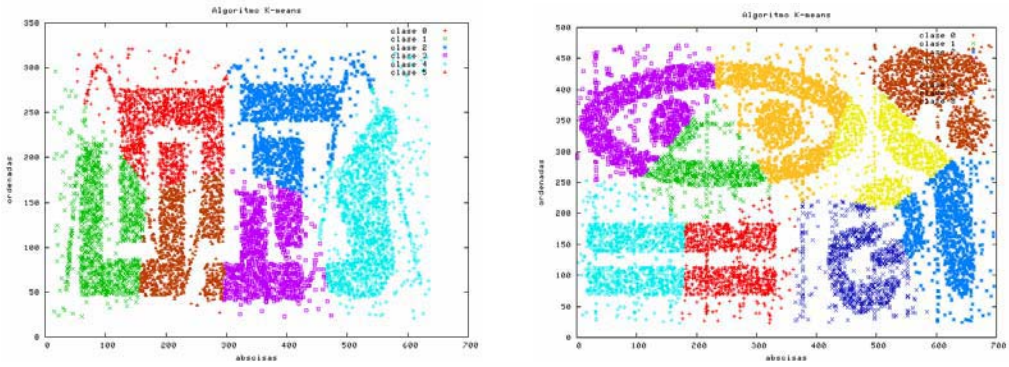


Figure 5. Results of the K-means algorithm on databases from [6]. Left: for 6 clusters. Right for 9 clusters.

3.3 Real Databases

Two real databases were used in this experiment, Iris and Cancer. The first one is a database of Iris plants containing 3 classes, with a total of 150 elements, 50 each of the three classes: Iris Setosa, Iris Versicolour, Iris Virginica. The number of attributes is 4, all numeric. The first class, Iris Setosa, is linearly separated from the other two.

In the first experiment, all the algorithms were run to obtain two classes, and all of them obtained 100% of correct grouping or classification, that is, all the tested algorithms were able to correctly separate the Setosa class from the other ones.

In a second experiment, the algorithms were run to find three clusters. The results are shown in Table 2. Notice how, due to the overlapping between Versicolour and Virginica classes, the proposed H-density algorithm outperforms the other ones reaching a 94% correct classification. In the case of the Cancer database, it has 2 classes. The proposed H-density algorithm obtained a 95.461% of correct classification, the same as CURE (Table 3).

Table 2. Classification rate of the clustering algorithm on Iris database.

Algorithm	% in two classes	% in three classes
DBSCAN	100	71.33
CURE	100	83.33
K-means	100	88.33
H-Density	100	94.00

Table 3. Classification rate of the clustering algorithms in Cancer database (two classes).

Database	DBSCAN	CURE	K-means	H-Density
Cancer	94.28	95.461	95.04	95.461

5 Conclusions and Further Work.

A hierarchical algorithm based on local probability density information has been presented. The way the density of the probability distribution is estimated, and the use of this information in the introduced dissimilarity measure between clusters, provides to the algorithm a mechanism to deal with overlapping distributions and the presence of noise in the data set. The experiments carried out show satisfactory and promising results to tackle these problems usually present in real databases. The experiments also show the proposed algorithm outperforms some existing algorithms. Future work is directed to unify the treatment of noise and overlapping in the process, and to introduce a measure to assess the right number of clusters in the hierarchy.

References

[1] Ester, M.; Kriegel, H. P.; Sander, J. and Xu, X.; A density-based algorithm for discovering clusters in large spatial databases with noise. In Proc. of the second International Conference on Knowledge Discovery and Data Mining, Portland, (1996) 226-231.

[2] Ertöz, L.; Steinbach, M. and Kumar V.: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. In Proceedings of Third SIAM International Conference on Data Mining, (2003).

[3] Fred A. L. and Leitao J.: A New Cluster Isolation Criterion Based on Dissimilarity Increments. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 25, No 8, (2003) 944-958.

[4] Guha, S.; Rastogi, R. and Shim, K.; CURE: An Efficient Clustering Algorithm for Large Databases. In Proceedings of ACM SIGMOD International Conference on Management of Data, . ACM, New York, (1998) 73-84.

[5] Hinneburg A. and Keim D.A.: An efficient Approach to Clustering in Large Multimedia Databases with Noise. In Proc. of the ACM SIGKDD, (1998).

[6] Karypis, G.; Han, E.H. and Kumar, V.; Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. In the IEEE Computer Society. Vol 32, No 8 (1999) 68-75.

[7] Tran T. N., Wehrens R. and Buydens L.M.C.: Knn Density-Based Clustering for High Dimensional Multispectral Images. Analytica Chimica Acta 490 (2003) 303–312.

Left/Right Deterministic Linear Languages Identification*

Jorge Calera-Rubio and Jose Oncina
Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos.
Apt.99. E-03080 Alicante, Spain.
{calera, oncina}@dlsi.ua.es

Abstract

Left deterministic linear languages are a subclass of context free languages that includes all regular languages. Recently was proposed an algorithm to identify in the limit with polynomial time and data such class of languages. It was also pointed that a symmetric class, right deterministic linear languages, is also identifiable in the limit from polynomial time and data. In this paper we show that the class of the Left-Right Deterministic Languages formed by the union of both classes is also identifiable. The resulting class is the largest one for which this type of results has been obtained so far.

In this paper we introduce the notion of n-negative characteristic sample, that is a sample that forces an inference algorithm to output a hypothesis of size bigger than n when strings from a non identifiable language are provided.

Keywords: example-based learning, learning context-free languages

1 Introduction

Over the time, diverse paradigms has been proposed to formalize when a learning process is successful. One of those paradigms is the *identification in the limit* [1]. In the identification in the limit paradigm, the learning process is seen as an infinite process in which an algorithm receives items of information about a target model. Each time an item is received the algorithm should propose a hypothesis. In order to be successful the algorithm should assure that, after receiving a finite number of items, the hypothesis model is always equivalent to the target.

Unfortunately, this paradigm does not put any restriction on the resources the inference algorithm can use. Several criteria has been introduced to cover this gap [2]. In this paper we are going to use the criterion of *identification in the limit from polynomial time and data* introduced by de la Higuera [3]. This criterion requires the

*Work partially supported by Spanish CICyT though project TIC2003-08496-C04 and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

existence of a polynomial size set of items such that, when provided to the algorithm along with other items, the algorithm should produce a hypothesis equivalent to the target.

In our case the models are formal languages and the items strings with labels indicating their belonging or not to the grammar.

The class of the *Linear Languages* is a subclass of the *Context Free* Grammars. This languages are produced by Context Free Grammars such that on the right hand side of the rules there is at most a nonterminal. Although it seems a quite simple class, it has been shown that even the question of saying if two linear languages are equivalent is undecidable. Then, the identification in the limit from polynomial time and data is impossible.

An important subclass of the linear languages are the *Left Deterministic Linear Languages* (LDLL) [4]. Those languages can be generated by *Left Deterministic Linear Grammars* that shares with the Deterministic Regular Grammar the property of knowing which is the next rule to use in the parsing of a string just by observing the leftmost terminal in the unparsed part of the string. $\{a^n b^n | n \geq 0\}$ and $\{a^m b^n c^n | m, n \geq 0\}$ are some examples of languages in the class, while $\{a^n b^n c^m | m, n \geq 0\}$ is not. This class includes the Regular Languages.

Unfortunately this class is not closed over the reversibility, that is not every language formed by the reversals of the strings in an LDLL are in LDLL. The class formed by the reversals of the languages in LDLL is called the *Right Deterministic Linear Languages* (RDLL) and are defined in a symmetrical way in which LDLL are defined. The regular languages, $\{a^n b^n | n \geq 0\}$ and $\{a^n b^n c^m | m, n \geq 0\}$ are examples of languages in the class.

Recently it was showed ([4]) that the class of the LDLL can be identified in the limit from polynomial time and data. Obviously, the class of the RDLL can also be identified just by reversing the strings before to introduce them on the LDLL inference algorithm.

In this paper we propose a new class of languages, the Left-Right Deterministic Languages (LRDLL), formed by the union of the LDLL and RDLL. We show that this class can be identified in the limit from polynomial time and data. The inference algorithm makes use of two inference algorithms, one for LDLL and other for RDLL. When a sample is given the main algorithm runs both inference algorithms and outputs the smaller of the hypothesis produced by the algorithms.

In order to show the identification in the limit from polynomial time and data, the concept of n-negative characteristic sample is introduced. That is a sample that forces an inference algorithm to output a hypothesis of size bigger than n when strings from a non identifiable language are provided.

The paper is organized as follows:

- Section 2 introduces the main notation used trough the paper, defines the classes of grammars object of this paper and defines the identification in the limit form polynomial time and data learning paradigm.
- Section 3 reviews the main properties of the LDLL and describes a learning algorithm for this class of languages.
- Section 4 describes the learning algorithm for the LRDLL, introduces the concept of n-negative characteristic set and it is used to show the identifiability in the limit from polynomial time and data.
- Section 5 concludes and propose new related open problems.

2 Definitions

2.1 Languages and Grammars

An alphabet Σ is a finite nonempty set of symbols. Σ^* denotes the set of all finite strings over Σ, $\Sigma^+ = \Sigma^* - \{\lambda\}$ where λ denotes the empty string. A language L over Σ is a subset of Σ^*. In the following, unless stated otherwise, symbols are indicated by $a, b, c \ldots$ and strings by $u, v, \ldots$. $\mathbb{N}$ is the set of non negative integers. The length of a string u will be denoted $|u|$, so $|\lambda| = 0$. Let I be a finite set of strings, $|I|$ denotes the number of strings in the set and $\|I\|$ denotes the total sum of the lengths of all strings in I.

Let $u, v \in \Sigma^*, u^{-1}v = w$ such that $v = uw$ (undefined if u is not a prefix of v) and $uv^{-1} = w$ such that $u = wv$ (undefined if v is not a suffix of u). Let L be a language and $u \in \Sigma^*$, $u^{-1}L = \{v : uv \in L\}$ and $Lu^{-1} = \{v : vu \in L\}$.

Let L be a language, the *prefix set* is $\Pr(L) = \{x : xy \in L\}$.The symmetrical difference between two languages L_1 and L_2 will be denoted $L_1 \ominus L_2$. The *longest common suffix* ($\mathrm{lcs}(L)$) of L is the longest string u such that $(Lu^{-1})u = L$.

Let u^R denote the reversal of the string u, the reversal of a string can be computed recursively as $(\lambda)^R = \lambda$ and $(ua)^R = au^R$. Let X be a set of strings $X^R = \{x^R : x \in X\}$.

Definition 1 (Context-free grammars). *A* context-free grammar *(CFG) G is a quadruple (Σ, V, P, S) where Σ is a finite alphabet (of terminal symbols), V is a finite alphabet (of variables or non-terminals), $P \subset V \times (\Sigma \cup V)^*$ is a finite set of production rules, and $S(\in V)$ is the axiom. We will denote $uTv \rightarrow uwv$ when $(T, w) \in P$.*

If there exists $u_0, \ldots, u_k$ such that $u_0 \rightarrow \cdots \rightarrow u_k$ we will write $u_0 \overset{k}{\rightarrow} u_k$. We denote by $L_G(T)$ the language $\{w \in \Sigma^* : T \overset{*}{\rightarrow} w\}$ and by $L(G)$ the language $\{w \in \Sigma^* : S \overset{*}{\rightarrow} w\}$. were $\overset{*}{\rightarrow}$ denotes the transitive, reflexive clousure of $\rightarrow$. Two grammars are equivalent if they generate the same language.

Let $G = (\Sigma, V, P, S)$ a CFG, the CFG $G^R = (\Sigma, V, P', S)$ is the reversal of G iff $(A, \alpha) \in P \iff (A, \alpha^R) \in P'$. Obviously, $x \in L(G) \iff x^R \in L(G^R)$.

Definition 2 (Linear grammars). *A context-free grammar* $G = (\Sigma, V, P, S)$ *is* linear *if* $P \subset V \times (\Sigma^* V \Sigma^* \cup \Sigma^*)$

We will be needing to speak of the *size* of a grammar. Without entering into a lengthy discussion, the size has to be a quantity polynomially linked with the number of bits needed to encode a grammar [2]. We will consider here the size of G denoted by $\|G\| = \sum_{(T,u)\in P}(|u| + 1)$.

2.2 Deterministic Linear Grammars

In [4] was introduced the class of the Left Deterministic Linear Grammars and Right Deterministic Linear Grammars as follows:

Definition 3 (Left Deterministic Linear Grammars). *A* Left Deterministic Linear Grammar *(LDLG)* $G = (\Sigma, V, P, S)$ *is a linear grammar where* $P \subset (V \times \Sigma V \Sigma^*) \cup (V \times \{\lambda\})$ *and*

$$\left.\begin{array}{l} \forall A \in V \\ \forall a \in \Sigma \\ \forall \alpha, \beta \in V\Sigma^* \end{array}\right. \quad \left.\begin{array}{l} (A, a\alpha) \in P \\ (A, a\beta) \in P \end{array}\right\} \Rightarrow \alpha = \beta$$

Definition 4 (Right Deterministic Linear Grammars). *A* Right Deterministic Linear Grammar *(RDLG)* $G = (\Sigma, V, P, S)$ *is a linear grammar where* $P \subset (V \times \Sigma^* V \Sigma) \cup (V \times \{\lambda\})$ *and*

$$\left.\begin{array}{l} \forall A \in V \\ \forall a \in \Sigma \\ \forall \alpha, \beta \in \Sigma^* V \end{array}\right. \quad \left.\begin{array}{l} (A, \alpha a) \in P \\ (A, \beta a) \in P \end{array}\right\} \Rightarrow \alpha = \beta$$

The languages generated by LDLG and RDLG are called Left Deterministic Linear Languages (LDLL) and Right Deterministic Linear Languages (RDLL) respectively.

Note that, RDLG = $(\text{LDLG})^R$ and then RDLL = $(\text{LDLL})^R$. The Deterministic Regular Grammars are a special case of the LDLG (RDLL) where the string that

appear on the rightmost (leftmost) part of the rules is λ. Then the LDLL and the RDLL include the regular languages. Both classes of languages include languages such as $\{a^n b^n | n \geq 0\}$ however, $\{a^m b^n c^n | m, n \geq 0\} \in$ LDLL but $\notin$ RDLL and, obviously, its reverse $\{c^n b^n a^m | m, n \geq 0\} \in$ RDLL but $\notin$ LDLL.

Finally, we are interested in deterministic linear grammars which can be LDLG or RDLG:

Definition 5 (Left-Right Deterministic Linear Grammars). *A CFG G is* Left-Right Deterministic Linear Grammar (LRDLG) *iff $G \in LDLG \cup RDLG$.*

The languages generated by LRDLG are called Left-Right Deterministic Linear Languages (LRDLL).

This class is closed over the reversal operation.

2.3 Learning and Identifying

In this paper we are concerned with the identification in the limit from polynomial time and data using positive and negative information. In this setting the learner is asked to learn from a learning *sample*, *i.e.* a finite set of strings, each string labelled by '+' if the string is a positive instance of the language (an element of L), or by '$-$' if it is a negative instance of the language (an element of $\Sigma^* - L$). Alternatively we denote $I = (I_+, I_-)$ where I_+ is the sub-sample of positive instances and I_- the sub-sample of negative ones.

Definition 6 (Identification in the limit from polynomial time and data). *A class $\mathcal{L}$ of languages is identifiable in the limit from polynomial time and data in terms of a grammar class $\mathcal{G}$* iff *there exist two polynomials $p()$ and $q()$ and an inference algorithm $\phi(\cdot)$ such that:*

1. *Given any sample (I_+, I_-), $\phi(I)$ returns a grammar $G \in \mathcal{G}$ such that $I_+ \subseteq L(G)$ and $I_- \cap L(G) = \emptyset$ in $O(p(\|I\|))$ time;*
2. *$\forall L \in \mathcal{L}$ and $\forall G \in \mathcal{G} : L(G) = L$, there exists a sample $C = (C_+, C_-)$(called characteristic) such that $\|C\| < q(\|G\|)$ for which, if $C_+ \subseteq I_+$, $C_- \subseteq I_-$, $\phi(I)$ returns a grammar G' such that $L(G') = L$.*

To simplify, we are going to say that a class of grammars $\mathcal{G}$ is identifiable in the limit from polynomial time and data if the class of languages $\mathcal{L} = L(\mathcal{G})$ is identifiable in the limit from polynomial time and data in terms of $\mathcal{G}$.

With this definition it is known that deterministic finite automata [5] and even linear grammars [6] are identifiable in the limit from polynomial time and data whereas non-deterministic finite automata and linear (and hence context-free) grammars are not [3].

3 Left Deterministic Linear Languages

As was pointed in section 2.2, the definition of LDLL is somewhat similar to the definition of Deterministic Regular Grammars. This similarity is going to allow us to define a normal form and a canonical automaton for such type of languages.

3.1 Canonical form

Let us first define the common suffix free languages that are going to play the role of the set of tails in a regular language.

LDLL use common suffix properties; in the sequel we are going to denote the *longest common suffix reduction* of a language L by $L\downarrow = L(\mathrm{lcs}(L))^{-1}$.

Definition 7 (Common suffix-free language equivalence). *Given a language L we define recursively the* common suffix-free *languages* $\mathrm{CSF}_L(\cdot)$*, and the associated equivalence relation as follows:*

$$\begin{array}{l} \mathrm{CSF}_L(\lambda) = L \\ \mathrm{CSF}_L(xa) = (a^{-1}\,\mathrm{CSF}_L(x))\downarrow \end{array} \quad \Bigg| \quad x \equiv_L y \iff \mathrm{CSF}_L(x) = \mathrm{CSF}_L(y)$$

It was shown in [4] that, a $L \in$ LDLL iff $\{\mathrm{CSF}_L(x) : x \in \Sigma^*\}$ is finite.

A consequence of this is the following corolary:

Corollary 1. *Let $L \notin$ LDLL then $|\{\mathrm{CSF}_L(x) : x \in \Sigma^*\}| = \infty$*

Now, following the parallelism with the Deterministic Regular Grammars, the canonical grammar for a LDLL can be defined as follows:

Definition 8 (canonical grammar for LDLL). *Given any linear deterministic language L, the associated canonical grammar is $G_L = (\Sigma, V, P, S_{\mathrm{CSF}_L(\lambda)})$ where:*

$$\begin{aligned} V =& \{S_{\mathrm{CSF}_L(x)} : \mathrm{CSF}_L(x) \neq \emptyset\} \\ P =& \{S_{\mathrm{CSF}_L(x)} \to a S_{\mathrm{CSF}_L(xa)}\,\mathrm{lcs}(a^{-1}\,\mathrm{CSF}_L(x)) : \mathrm{CSF}_L(xa) \neq \emptyset\} \\ & \cup \{S_{\mathrm{CSF}_L(x)} \to \lambda : \lambda \in \mathrm{CSF}_L(x)\} \end{aligned}$$

We are going to define now a cannonical form to write this grammar:

Definition 9 (Advanced form for LDLL). *A linear grammar $G = (\Sigma, V, P, S)$ is deterministic in* advanced *form if:*

1. *all rules are in the form $(T, aT'w)$ or (T, λ);*

2. $\forall (T, aT'w) \in P, w = \mathrm{lcs}(a^{-1}L_G(T))$;
3. *all non-terminal symbols are accessible:* $\forall T \in V\ \exists u, v \in \Sigma^* : S \xrightarrow{*} uTv$ *and useful:* $\forall T \in V,\ L_G(T) \neq \emptyset$;
4. $\forall T, T' \in V,\ L_G(T) = L_G(T') \Rightarrow T = T'$.

Now it was proved in [4] that:

Theorem 1. *Let $L \in$ LDLL, then G_L is the smallest LDLL advanced grammar such that $L(G_L) = L$. Moreover, it is unique up to isomorphisms.*

3.2 Learning LDLL

As LDLL admit a small canonical form it is sufficient to have an algorithm that can learn this type of canonical form at least when a characteristic set is provided. In doing so we are following the type of proof used to prove learnability of *dfa* [7, 5].

The idea of the algorithm is to provide a systematic way to build the canonical grammar provided we can make some type of queries to an unlimited oracle. In a second step, the queries to the oracle are changed by functions that extract equivalent information from the learning set.

Let first introduce the concept of *short prefix*:

Definition 10. *Let L be a LDLL, and $\leqslant$ a length lexicographic order relation over Σ^*, the* shortest prefix *set of L is defined as* $\mathrm{Sp}_L = \{x \in \mathrm{Pr}(L) : \mathrm{CSF}_L(x) \neq \emptyset \wedge y \equiv_L x \Rightarrow x \leqslant y\}$

Note that, in a canonical grammar, we have a one to one relation between strings in Sp and non terminals of the grammar. We shall thus use the strings in Sp as identifiers for the non terminal symbols.

Imagine we have an unlimited oracle that knows language L and to which we can address the following queries:

$$\mathrm{next}(x) = \{xa : \exists xay \in L \wedge \mathrm{CSF}_L(xa) \neq \emptyset\} \qquad \mathrm{equiv}(x, y) \iff x \equiv_L y$$
$$\mathrm{right}(xa) = \mathrm{lcs}(a^{-1}\,\mathrm{CSF}_L(x)) \qquad \mathrm{isfinal}(x) \iff \lambda \in \mathrm{CSF}_L(x)$$

An algorithm (alg. 1) can be built to construct the canonical grammar. Algorithm 1 visits the prefixes of the language L in length lexicographic order, and constructs the canonical grammar responding to definition 8. If a prefix xa is visited and no previous equivalent non terminal has been found (and placed in Sp), this prefix is added to Sp as a new non terminal and the corresponding rule is added to the grammar. If there exists an equivalent non terminal y in Sp then the corresponding rule is added but the strings for which x is a prefix will not be visited (they will not

be added to W). When the algorithm finishes, Sp contains all the short prefixes of the language.

In order to simplify notations we introduce:

Definition 11.

$$\forall x : \mathrm{CSF}_L(x) \neq \emptyset,\ \mathrm{tail}_L(x) = \begin{cases} \mathrm{lcs}(x^{-1}L) & \text{if } x \neq \lambda \\ \lambda & \text{if } x = \lambda \end{cases}$$

Lemma 1. *Let $G_L = (\Sigma, V, P, S)$ be the canonical grammar of a LDLL L, $\forall x : CSF(x) \neq \emptyset$,*

1. $\mathrm{lcs}(a^{-1}\,\mathrm{CSF}_L(x)) = (\mathrm{tail}_L(xa))(\mathrm{tail}_L(x))^{-1}$
2. $xv\,\mathrm{tail}_L(x) \in L \iff v \in L_{G_L}([x])$.

In order to use algorithm 1 with a sample $I = (I_+, I_-)$ instead of an oracle with access to the whole language L the 4 functions next, right, equiv and isfinal have to be implemented as functions of $I = (I_+, I_-)$ rather than of L:

$$\begin{aligned} \mathrm{next}(x) &= \{xa : \exists xay \in I_+\} \\ \mathrm{right}(xa) &= \mathrm{tail}_{I_+}(xa)\,\mathrm{tail}_{I_+}(x)^{-1} \\ \mathrm{equiv}(x,y) &\iff xv\,\mathrm{tail}_{I_+}(x) \in I_+ \Rightarrow yv\,\mathrm{tail}_{I_+}(y) \notin I_- \\ &\quad \wedge yv\,\mathrm{tail}_{I_+}(y) \in I_+ \Rightarrow xv\,\mathrm{tail}_{I_+}(x) \notin I_- \\ \mathrm{isfinal}(x) &\iff x\,\mathrm{tail}_{I_+}(x) \in I_+ \end{aligned}$$

Algorithm 1 Computing G using functions next, right, equiv and isfinal

Require: functions next, right, equiv and isfinal, language L
Ensure: $L(G) = L$ with $G = (\Sigma, V, P, S_\lambda)$
  $\mathrm{Sp} = \{\lambda\}$; $V = \{S_\lambda\}$
  $W = \mathrm{next}(\lambda)$
  while $W \neq \emptyset$ **do**
    $xa = \min_{\leq} W$
    $W = W - \{xa\}$
    if $\exists y \in \mathrm{Sp} : \mathrm{equiv}(xa, y)$ **then**
      add $S_x \to aS_y\,\mathrm{right}(xa)$ to P
    else
      $\mathrm{Sp} = \mathrm{Sp} \cup \{xa\}$; $V = V \cup \{S_{xa}\}$
      $W = W \cup \mathrm{next}(xa)$
      add $S_x \to aS_{xa}\,\mathrm{right}(xa)$ to P
    end if
  end while
  for all $x \in \mathrm{Sp} : \mathrm{isfinal}(x)$ **do**
    add $S_x \to \lambda$ to P
  end for

It is easy to see that, if a set fulfils the following conditions, then the algorithm will be force to output the canonical grammar (see [4] for detail).

Definition 12 (characteristic sample). *Let $I = (I_+, I_-)$ be a sample of the LDLL L. I is a* characteristic sample *(CS) of L if:*

1. $\forall x \in \mathrm{Sp}_L \, \forall a \in \Sigma : xa \in \mathrm{Pr}(L) \Rightarrow \exists xaw \in I_+$
2. $\forall x \in \mathrm{Sp}_L \, \forall a \in \Sigma : \mathrm{CSF}_L(xa) \neq \emptyset \Rightarrow \mathrm{tail}_{I_+}(xa) = \mathrm{tail}_L(xa)$
3. $\forall x, y \in \mathrm{Sp}_L \, \forall a \in \Sigma : \mathrm{CSF}_L(xa) \neq \emptyset \wedge xa \not\equiv_L y \Rightarrow$
   $\exists v : xav \, \mathrm{tail}_L(xa) \in I_+ \wedge yv \, \mathrm{tail}_L(y) \in I_- \; \vee$
   $\exists v : yv \, \mathrm{tail}_L(y) \in I_+ \wedge xav \, \mathrm{tail}_L(xa) \in I_-$
4. $\forall x \in \mathrm{Sp}_L : x \, \mathrm{tail}_L(x) \in L \Rightarrow x \, \mathrm{tail}_L(x) \in I_+$

Condition 1 assures that all the non terminals will be represented on the output grammar. Condition 2 assures that the right hand part of the rules will be well constructed. Condition 3 assures that every non equivalent non terminals tested on the algorithm will be detected as non equivalent. And condition 4 assures that all the rules with shape $A \to \lambda$ will be included in the grammar.

In [4] was proved that a polynomial set that fulfils all the conditions can be build.

As a corollary of that we have:

Corollary 2. *The LDLG can be identified in the limit from polynomial time and data using positive and negative sample.*

4 Learning LRDLG

On the previous section we have defined an algorithm LDLGA($\cdot$) that identifies the LDLG. Reminding that RDLL = LDLL^R, then it is easy to build an algorithm RDLGA($\cdot$) for RDLG such that RDLGA(I) = (LDLGA(I^R))^R.

Now, for the LRDLG let us define an algorithm (LRDLGA) (see alg. 2) that given a sample, uses it with LDLGA and RDLGA, and returns the hypothesis grammar with a lower number of non terminals.

If the target language is in LDLL$-$RDLL and the sample is enough big, LDLGA will provide the canonical LDLG for the language, but the RDLGA, by corollary 1, is going to produce bigger and bigger grammars as the sample grows. Then it has to exist a point when the correct hypothesis will be outputted.

The case when the language is in RDLL$-$LDLL is similar. And the case when the target is in both classes, LRDLGA will output the smaller of both representations.

Now, in order to show the identification in the limit from polynomial time and data, we have to show the existence of a polynomial characteristic set. The idea is to find a sample such that, if the language is not in the class it will force the algorithm to output a hypothesis with size bigger that a given parameter. Let us formalize this idea:

Definition 13 (n-negative characteristic sample). *Let $\phi(\cdot)$ an inference algorithm that identifies in the limit the class of languages $\mathcal{L}$ in terms of the class of grammars $\mathcal{G}$, let $L \notin \mathcal{L}$, $C = (C_+, C_-)$ is a n-*negative characteristic sample *(n-NCS) for ϕ if for all sample $I = (I_+, I_-)$ of $L : C_+ \subseteq I_+, C_- \subseteq I_-$, then $\|\phi(I)\| \geq n$.*

In our case, we are going to use the number of non terminals as the size of a grammar. Then, if we can show that for every language in $\mathrm{LDLL} - \mathrm{RDLL}$ (or $\mathrm{RDLL} - \mathrm{LDLL}$) with n non terminals we can find a polynomial size $(n+1)$-NCS for RDLGA (LDLGA), the union of the characteristic sample for LDLGA (RDLGA) of the language with the $(n+1)$-NCS will be a polynomial size characteristic sample for the LRDLGA.

Let us show that this polynomial size n-NCS exists.

Proposition 1. *Let $L \notin LDLL$ and let $n \in \mathbb{N}$. As $L \notin LDLL$ we know that $|\,\mathrm{Sp}_L\,|$ is infinite, let Sp_{L_n} be the set of n smallest elements $x \in \mathrm{Sp}_L$ in the lenght lexicographic order. Let $I = (I_+, I_-)$ be a sample of L, I is an n-negative characteristic sample (n-NCS) for LRDLGA if:*

1. $\forall x \in \mathrm{Sp}_{L_n}\, \forall a \in \Sigma : xa \in \Pr(L) \Rightarrow \exists xaw \in I_+$
2. $\forall x \in \mathrm{Sp}_{L_n}\, \forall a \in \Sigma : \mathrm{CSF}_L(xa) \neq \emptyset \Rightarrow \mathrm{tail}_{I_+}(xa) = \mathrm{tail}_L(xa)$
3. $\forall x, y \in \mathrm{Sp}_{L_n}\, \forall a \in \Sigma : \mathrm{CSF}_L(xa) \neq \emptyset \wedge xa \not\equiv_L y \Rightarrow$
   $\exists v : xav\,\mathrm{tail}_L(xa) \in I_+ \wedge yv\,\mathrm{tail}_L(y) \in I_- \;\vee$
   $\exists v : yv\,\mathrm{tail}_L(y) \in I_+ \wedge xav\,\mathrm{tail}_L(xa) \in I_-$
4. $\forall x \in \mathrm{Sp}_{L_n} : x\,\mathrm{tail}_L(x) \in L \Rightarrow x\,\mathrm{tail}_L(x) \in I_+$

Algorithm 2 Computing the grammar G for a language $L \in \mathrm{LRDLL}$

Require: Algorithm 1, language L
Ensure: $L(G) = L$ with $G = (\Sigma, V, P, S_\lambda)$ and $|V|$ smaller.
  Let $G_L = (\Sigma, V_L, P_L, S_{\lambda,L})$ the grammar computed by algorithm 1 with L as input.
  Let $G_R = (\Sigma, V_R, P_R, S_{\lambda,R})$ the reversed grammar computed by algorithm 1 with L^R as input.
  if $|V_L| \leq |V_R|$ **then**
    $G = G_L$
  else
    $G = G_R$
  end if

Proof. As $L \notin$ LDLL we know that $\{CSF_L(x)\}$ is infinite and then, if we try to build a canonical grammar, we are going to obtain an infinite number of non terminals. Observe that the inference algorithm constructs the grammar iteratively from the non terminal nearest to the start symbol to the farthest. The conditions of the n-NCS provide enough information to the inference algorithm to construct correctly the productions related to the first n non terminal of the infinite grammar.

It is easy to see that condition 1 assures that all the first n non terminals will be represented on the output grammar. Condition 2 assures that the right part of the rules related with the first n non terminals will be well constructed. Condition 3 assures that every non equivalent non terminals (of the first n) tested on the algorithm will be detected as non equivalent. And condition 4 assures that all the rules with shape $A \to \lambda$, for the first n non terminals will be included in the grammar. □

Now we have to show that there is a polynomial sample that fulfils the previous conditions. In order to show this, the following lemma proved in [4] is needed.

Lemma 2. *Let $G_L = (\Sigma, V, P, S)$ be the canonical grammar of a LDLL L, and let x, y be such that $\mathrm{CSF}_L(x) \neq \mathrm{CSF}_L(y)$, then $\exists z \in L_{G_L}([x]) \ominus L_{G_L}([y])$ such that $|z| \leq ||G_L||^2$.*

Theorem 2. *For any $L \notin$ LDLL there is an n-negative characteristic sample of polynomial size.*

Proof. Obviously, the number of strings involved in the conditions is polynomial, then we have to show that their lengths are also polynomial.

Note that each time a production rule of a LDLG is applied in the parsing of a string, a non terminal is removed from the prefix of a string, then $\forall x \in \mathrm{Sp}_{L_n} : |x| \leq n$. Those strings needs to reach the non terminal represented by the short prefix x use the rule whose right hand part beginning with terminal a and then a string to reach a final non terminal (a terminal A such that $A \to \lambda \in P$). So, the length of the strings xaw in condition 1 of proposition 1 are bounded by $(2n+1)(|w_l|+1)$, where w_l is the longest suffix in the right hand side of the production rules of G.

In a similar way, we can see that the length of strings related with condition 2 and 4 can also be bounded by $(2n+1)(|w_l|+1)$.

Finally, lemma 2 shows that the length of the strings necessary for third condition can be quadratically bounded. □

Example 1. Consider the language $L = \{a^n b^n c^m : n \geq 0, m \geq 0\}$. This language is in RDLL but is not in LDLL. The right canonical grammar G_R for it is:

$$\begin{aligned} S &\longrightarrow Sc \\ S &\longrightarrow aAb \\ S &\longrightarrow \lambda \\ A &\longrightarrow aAb \\ A &\longrightarrow \lambda \end{aligned}$$

We are going to show that the left canonical grammar has an infinite number of non terminals.

The characteristic sample (I_+, I_-) with $I_+ = \{\lambda, c, ab, abc, aabb, abcc\}$ and $I_- = \{acb, aaccbb\}$ let us identify[1] G_R with $\mathrm{Sp}_R = \{\lambda, a\}$, ($S_\lambda \equiv S$, $S_a \equiv A$). On the other hand, we can compute $CSF_L(x)$ for every $x \in \mathrm{Pr}(L)$:

x	$CSF(x)$
λ	$a^n b^n c^m$
a	$a^n b^{n+1} c^m$
c	$a^n b^n c^m$
aa	$a^n b^{n+2} c^m$
ab	c^m
aaa	$a^n b^{n+3} c^m$
aab	bc^m
abc	c^m
$\cdots$	$\cdots$

One can see that in this case Sp_L remains unbounded and, in order to identify correctly the grammar as RDLG with algorithm LRDLGA, is sufficient supply a 3-NCS. If we add the string $aabbc$ to I_+ and the strings b and bb to I_-, the sample provided above becomes 3-NCS for languages in LDLL, giving the left grammar:

$$\begin{aligned} S_\lambda &\longrightarrow aS_a \\ S_\lambda &\longrightarrow cS_\lambda \\ S_\lambda &\longrightarrow \lambda \\ S_a &\longrightarrow aS_{aa} \\ S_a &\longrightarrow bS_\lambda \\ S_{aa} &\longrightarrow bS_a \end{aligned}$$

[1] Recall that the input to algorithm 1 must be the reversal of the sample, and the obtained grammar must be also reversed.

5 Summary and Future Work

Left Deterministic Linear Languages (LDLL) and Right Deterministic Linear Languages (RDLL) are subclasses of Linear Languages that, in turn, includes the Regular Languages. We define the Left-Right Deterministic Languages (LRDLL) as the union of the LDLL and RDLL.

In this paper we have proved that the class of the LRDLL is identifiable in the limit from polynomial time and data. This class of languages is the largest one for which this type of results has been obtained so far. To do so we have introduced the notion of n-negative characteristic sample as a sample that forces an inference algorithm to produce a hypothesis of size n when strings from a non identifiable grammar are provided.

Note that in the parsing of a string by a LDLG if we have reached a non terminal, the next rule to apply can be determined by looking the leftmost terminal of the non parsed string. In RDLG the nonterminal to look is the rightmost. Let we call the non terminals in LDLG left deterministic while the non terminals in RDLG right deterministic.

Now, a new class of linear languages can be defined as a grammars such that each non terminal is left deterministic or right deterministic, but not both at the same time. It is easy to see that, on such grammars, the parsing can be done in a deterministic way provided we know if the reached non terminal is left or right deterministic. This class includes properly the LRDLL.

Can the n-negative characteristic sample technique be expanded in order to elucidate if a non terminal is left or right deterministic? Can this class of grammars be identified from polynomial time and data?

6 Acknowledgement

The authors thank Colin de la Higuera for fruitful discussions on the subject.

References

[1] E.M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[2] L. Pitt. Inductive inference, DFA's, and computational complexity. In *Analogical and Inductive Inference*, number 397 in Lecture Notes in Artificial Intelligence, pages 18–44. Springer-Verlag, Berlin, 1989.

[3] C. de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27:125–138, 1997.

[4] C. de la Higuera and J. Oncina. Learning deterministic linear languages. In *Computational Learning Theory, COLT 02*, number 2375 in Lecture Notes in Artificial Intelligence, pages 185–200. Springer Verlag, 2002.

[5] J. Oncina and P. García. Identifying regular languages in polynomial time. In H. Bunke, editor, *Advances in Structural and Syntactic Pattern Recognition*, volume 5 of *Series in Machine Perception and Artificial Intelligence*, pages 99–108. World Scientific, 1992.

[6] J.M. Sempere and P. García. A characterisation of even linear languages and its application to the learning problem. In *Grammatical Inference and Applications, ICGI'94*, number 862 in Lecture Notes in Artificial Intelligence, pages 38–44. Springer Verlag, 1994.

[7] E.M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37:302–320, 1978.

Band selection using mutual information matrix for hyperspectral data

J.M. Sotoca, F. Pla
Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, E-12071 Castelló de la Plana (Spain)
E-mail: {sotoca,pla}@uji.es

Abstract

In this paper, a band selection technique for hyperspectral image data is proposed. A mutual information matrix between pairs of bands is built to collect the relations of information between the different regions of the spectrum. A process based on a Deterministic Annealing optimization is applied on the mutual information matrix to obtain a probabilistic model and look for the image bands less uncorrelated as possible between them. Two supervised filter feature selection methods were also tested to analyze the accuracy obtained by the presented approach. The proposed methodology can develop for supervised selection, building the matrix in terms of class separability for labelled training sets.

1 Introduction

Hyperspectral sensors acquire information in large quantities of spectral bands, which generate hyperspectral data in high dimensional spaces. These systems use spectral information to perform certain tasks in remote sensing, medical imaging, product quality assessment, and so on. These systems use multispectral image representations in order to estimate and analyze the presence of vegetation pathologies, substances or chemical compounds, pathologies, etc, providing a qualitative and quantitative evaluation of those features.

A multispectral image can be considered as defined in a $3D$ space $I(x, y, \lambda)$, where (x, y) denotes the spatial co-ordinates of the pixel location in the image, and λ denotes a spectral band (wavelength). Each spectral band records a specific portion of the electromagnetic spectrum so that each spectral band provides greater insight about the composition of the different regions of the image. Therefore, each image band is captured at the selected wavelength with a narrow band-pass filter, allowing a multi-band representation.

When having available hyperspectral data, a common question to be solved is how to select the right spectral bands to characterize the problem. The main objective of band selection in multispectral imaging is to avoid redundant information and reduce the amount of data to be processed. Therefore, from the point of view of remote sensing, we would

Edited by F.Pla, P.Radeva, J.Vitrià, 2006.

be interested in feature selection [8] rather than in feature extraction [10, 11]. For instance, obtaining a new set of reduced image representations from a linear combination of the whole set of original image bands is not desirable, since we would need the total amount of information to obtain the new features. On the other hand, selecting a subset of relevant bands from the original set, allows the process of image acquisition to be reduced to a certain number of bands instead of dealing with the whole amount of data, making simpler the image acquisition and analysis.

In the framework of multispectral imaging, another possible answer to the problem of feature selection would be using an unsupervised approach. One way to solve it consists of grouping the data in the feature space by using clustering techniques [2]. Another approach is to minimize the classification error by selecting bands that provide the highest image contrast [5]. In this work, a Deterministic Annealing (DA) approach is used to analyze the amount of information contained in the *mutual information matrix*, which represents the relations of information for pairs of spectral bands. The proposed algorithm uses a Deterministic Annealing (DA) approach to look for groups of bands as less correlated as possible, representing correlation between image bands by means of mutual information. Selected bands are further used in pixel classification tasks to assess the performance of proposed technique.

2 Transinformation matrix

Let us consider a pair of random variables A_i and A_j, representing the image bands i and j. The amount of information contained in both images can be expressed as the joint entropy $H(A_i, A_j)$, that is,

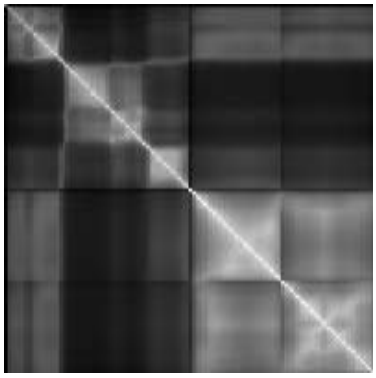
$$H(A_i, A_j) = \sum p(a_i, a_j) \log_2 \frac{1}{p(a_i, a_j)} \tag{1}$$

where $p(a_i, a_j)$ represents a joint probability distribution. The term $\log_2 \frac{1}{p(a_i,a_j)}$ means that the amount of information gained from a event with probability $p(a_i, a_j)$ is inversely related to the probability that this event take place. The rarer is an event, the more meaning is assigned to occurrence of the event. Thus, the information per event is weighted by the probability of occurrence. The resulting entropy term is the average amount of information gained from a set of possible events.

For two images i and j, the co-joint probability distribution $p(a_i, a_j)$ of both images can be estimated as,

$$p(a_i, a_j) = \frac{h(a_i, a_j)}{MN} \tag{2}$$

where $h(a_i, a_j)$ is the joint gray level histogram of both images, and the normalizing factor, MN (M columns and N rows) is the image size, assuming all images bands with equal size.



(a)

Figure 1: The Mutual Information matrix for a multispectral image with 128 wavebands. Darker values represent less correlation.

Mutual information $H(A_i{:}A_j)$ is a basic concept in information theory [1]. It measures the interdependence between random variables. In the case of two images, the mutual information is defined as:

$$H(A_i : A_j) = H(A_i) + H(A_j) - H(A_i, A_j) \tag{3}$$

where $H(A_i)$, $H(A_j)$ are the entropy of images i and j. The function $H(A_i{:}A_j)$ measures the amount of information shared between A_i and A_j. The entropies of both images satisfy the following inequality:

$$0 \leq H(A_i : A_j) \leq \min\{H(A_i), H(A_j)\} \tag{4}$$

One way to establish the interdependence between a set of features is defining the *transinformation matrix* (see Fig 1). This is a square matrix representing the mutual information between pairs of image bands. The diagonal terms represent the entropy of single band.

3 A new technique for rank reduction

We look for a strategy based on an unsupervised approach because, in supervised methods, it is necessary to fix beforehand the number of classes or regions present in the image, and to label the adequate number of training instances. Moreover, the computational cost of filter methods in supervised feature selection is considerable and, in many problems, labelling data can become a complex and difficult task.

Consider as input space the *transinformation matrix* with range D (number of spectral bands), representing the dependence among image bands. Contiguous bands in the spectrum tend to be highly correlated (brighter values in Fig 1). Looking at the *transinformation matrix*, we could interpret the problem of band selection as a rank reduction process of that matrix.

One possibility could be, for instance, to apply Truncate Singular Decomposition Value (TSVD) over the *transinformation matrix* or other factorization methods, eliminating the smaller singular values and their corresponding singular vectors. This idea has been used for noise reduction in signal processing [4].

The technique here proposed is aimed at reducing the rank of the *transinformation matrix* by selecting a given number of features that minimize the correlation among them. Therefore, we look for a global minimum without carrying out a search of subsets of features in the feature space. The process must be capable of picking up a few subset of bands in the mains regions that appear in the *transinformation matrix*, and obtaining as better performance as possible from the classification point of view reducing the feature space.

Given a certain function of information I_{ij} between pairs of bands represented in the matrix, we are interested in associating a probability of significance $p(I_{ij}|ij)$ for each position i and j in the matrix. This probability will mean how relevant is the interaction of band i and j for the problem. In the case of the *transinformation matrix*, each entry I_{ij} can represent the mutual information between bands.

On the other hand, discretizing I_{ij} values and representing them as gray levels (see Fig 1), allows to define a spreading measure of the information in the gray level distribution of the *transinformation matrix*. This measure will estimate the information contained about the appearance of the different regions of the spectrum in the matrix. Thus, we can consider the matrix as an "image" and analyze the probability that the event (value associate with each position of the matrix) take place. That is, the probability distribution associated to each position of the matrix n_{ij} can be calculated as $n_{ij} = h_{ij}/D^2$, where h_{ij} is the value in the histogram for the gray level at i and j.

Therefore, a probabilistic model is applied over each position of the matrix $p(I_{ij}|ij)$. It is, thus, possible to utilize DA to obtain the image bands that contain higher values of significance in the matrix. To apply DA in such a framework, the following requirements must be fullfiled:

- The entropy S of the distribution of probabilities $p(I_{ij}|ij)$ associated to this representation of "level of uncertainly" must be maximum.
- The sum of probabilities are normalized to one.
- The product of $p(I_{ij}|ij)$ per the value of I_{ij} between pairs of bands, provides a value about the amount of information I associated to the ensemble.

Therefore, we can establish the the following relation:

$$S = -\sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) \log \frac{p(I_{ij}|ij)}{p_{ij}} \tag{5}$$

subject to

$$\sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) = 1 \quad \text{and} \quad \sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) I_{ij} = I \tag{6}$$

where p_{ij} is proportional to the prior contribution of each relation between pairs of bands. Thus, S is the entropy relative to some "measures" p_{ij} that has to be maximized [6]. To maximize S subject to the constraint Eq 6, we can introduce Lagrangian multipliers α and β,

$$S + \alpha \sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) + \beta \sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) I_{ij} \tag{7}$$

Setting the partial derivative of Eq 7 with respect $p(I_{ij}|ij)$ to zero, we obtain the following expression,

$$-\log p(I_{ij}|ij) - 1 + \log p_{ij} + \alpha + \beta I_{ij} = 0 \tag{8}$$

where

$$p(I_{ij}|ij) = p_{ij} e^{\alpha - 1 + \beta I_{ij}} \tag{9}$$

Taking into account that the sum of probabilities are normalized to one, then

$$\sum_{i=1}^{D}\sum_{j=1}^{D} p_{ij} e^{\beta I_{ij}} = e^{1-\alpha} = Z \tag{10}$$

where Z is the so-called the *partition function* and

$$p(I_{ij}|ij) = \frac{p_{ij} e^{\beta I_{ij}}}{Z} \tag{11}$$

On the other hand, we have to fix the Lagrangian multiplier β such as I and S are related. This optimization can be conveniently reformulated as the minimization of the following Lagrangian F with a parameter T:

$$F = I - TS \tag{12}$$

Therefore, the corresponding minimum of F is obtained by putting the Eq 11 into Eq 12

$$F^* = \min(F) = -T \log \left(\sum_{i=1}^{D}\sum_{j=1}^{D} p_{ij} e^{\beta I_{ij}} \right) \tag{13}$$

Multiplying $p(I_{ij}|ij)$ in Eq 8 and adding for all values, we obtain

$$-\sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij)\log\frac{p(I_{ij}|ij)}{p_{ij}} - (1-\alpha)\sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij) + \beta\sum_{i=1}^{D}\sum_{j=1}^{D} p(I_{ij}|ij)I_{ij} = 0 \tag{14}$$

then

$$S + \beta I = 1 - \alpha = lnZ \tag{15}$$

and from the Eq 12

$$S - \frac{I}{T} = -\frac{F}{T} \tag{16}$$

Thus, we can consider $\beta = -1/T$ and $lnZ = -F/T$. Finally, our probability function is expressed as

$$p(I_{ij}|ij) = \frac{p_{ij}e^{-I_{ij}/T}}{\sum_{i=1}^{D}\sum_{j=1}^{D} p_{ij}e^{-I_{ij}/T}}$$

and

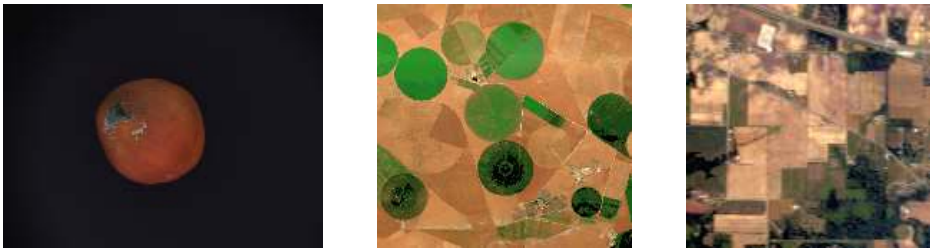
$$p_{ij} = I_{ij}p(I_{ij}|ij)$$

The result is the Bayes' Theorem, where we can obtain the posterior probability distribution for each position through the exponential function of the values observed in the matrix per the prior probability p_{ij}.

In our experiments, we have observed that using $I_{ij} = H(A_i{:}A_j)$, the approach finds a global minimum in regions of the spectrum of image bands with smaller values of mutual information with respect to the rest of regions of the spectrum represented in the *transinformation matrix*. Nevertheless, to obtain a good performance of the classifier in the subset of features selected, it is necessary to choose image bands of the different regions more representative of the ensemble. This question can be solved introducing the probability to appear this event in the matrix n_{ij} in the function of information as:

$$I_{ij} = n_{ij}H(A_i : A_j). \tag{17}$$

The initialization of DA starts with large enough values of T, and a uniform distribution of probabilities $p(I_{ij}|ij) = 1/D^2$. The initial set of features X to choose is empty. It is clear from Eq 12 that the goal at each temperature is to maximize the entropy of the partition. As $T \to 0$ a reduction of the amount of information I is carried out. In practice, the system is annealed to a low temperature, such the amount of information I ("level of dependence" of the matrix) is sufficiently small.



(a) (b) (c)

Figure 2: (a) Example of RGB composition for an orange image in the Visible spectrum. (b) HyMap RGB composition, Barrax, Spain. (c) RGB composition of AVIRIS (92AV3C: NW Indiana's Indian Pine test site).

On the other hand, we express the probability contributions of each band A_i accumulating for each row or column i (the matrix is symmetrical) as:

$$B_i = \sum_{j=1}^{D} p(I_{ij}|ij) \tag{18}$$

While T decreases, the difference between the values of $p(I_{ij}|ij)$ grow up. As T goes down, the probability contributions of some bands $B_i \rightarrow 0$, but it is possible that further in the annealing with lower T, previous low values of B_i grow up for the new circumstances. Only if $B_i \cong 0$, we can almost assure that the corresponding band will not contribute in the probability distribution in the next iterations.

Summarizing, a brief sketch of the algorithm is as follows:

1. *Initialize:* $T = T_0$, $p(I_{ij}|ij) = 1/D^2$ and $|X| = 0$
2. *Minimize:* $F = I - TS$
3. *Calculate:* $B_i = \sum_{j=1}^{D} p(I_{ij}|ij)$
4. *If* $B_i \cong 0$ *then:* $X \leftarrow (X \cup A_i)$
5. *Count the number of image bands* R *such:* $B_i > 1/D$
6. *Lower Temperature:* $T \leftarrow q(T)$
7. *Go to step 2 while* $R \geq 2$

In our experiments, we used and exponential schedule to reduce T, $q(T) = \alpha T$, where $\alpha < 1$, but other annealing schedules are possible. At the end of the algorithm, the probability contributions B_i are concentrated in the two best bands with values about $\simeq 0.5$.

4 Experiments and results

In the experiments, we have used four sources of hyperspectral or multispectral data. The two first collections of multispectral images were obtained by an imaging spectrograph (Retiga-Ex, Opto-knowledge System Inc. Canada). The first one has a spectral range from 400 to 720 nanometers in the visible (VIS) obtaining a set of 33 spectral bands for each image. The second one has a spectral range from 650 to 1050 nanometers in the near infrared (NIR) obtaining a set of 41 spectral bands for each image. In both cases, the camera has a spectral resolution of 10 nanometers.

The image database consisted of forty multispectral images for the VIS and NIR, respectively, corresponding to orange fruits with different types of defects and skin variations on their surfaces (see Fig 2 (a)). In order to compare the performance of the approach here presented, different region of the oranges, including the background, were labelled in eight classes, obtaining 1463346 pixels from VIS and 1491888 pixels from NIR.

The third source of data corresponds to a spectral image (700 X 670 pixels) acquired with the 128-bands HyMap spectrometer during the DAISEX-99 campaign with six different classes were considered in the area (see Fig 2(b)) (http:/io.uv.es/projects/daisex/).

The fourth source of data corresponds to a spectral image (145 X 145 pixels) acquired with the AVIRIS data set with 220 bands collected in June 1992 over the Indian Pine Test site in Northwestern Indiana (see Fig 2 (c)). The data set is designated as *92AV3C*, and it has seventeen classes.(http:/dynamo.ecn.purdue.edu /∼biehl/MultiSpec)

In order to assess the performance of the method, a Nearest Neighbor (NN) classifier was used to classify pixels into the different classes. The performance of the NN classifier was considered as the validation criterion to compare the significance of the subsets of selected image bands obtained by the proposed approach and two supervised methods considered in the experiment carried out. To increase the statistical significance of the results, the average values over five random partitions were estimated.

4.1 Supervised criteria proposed

To analyze the accuracy of the ranking of bands obtained by the proposed approach, two supervised filter feature selection methods were also tested. Thus, the band selection process was considered as a supervised feature selection approach, in this case using the labelled data set for the feature selection process.

The main motivation about comparing the proposed method with supervised approaches is that the labelled data contains information about the distribution of classes exiting in the hyperspectral data, and they allow the search for relevant feature subsets. Comparing the performance with those approaches, we can measure the capability to obtain subsets of relevant features (image bands) by the introduced DA approach without a prior knowledge

of the class distributions in the multispectral image.

The first method is the well-known *ReliefF* algorithm [9] based on pattern distances. This algorithm initializes every feature weight to zero and then iterates m times looking for a set of feature weights that optimizes a criterion function.

The procedure begins by randomly selecting a sample x from the data set. For the selected sample, it determines the nearest neighbor prototype of the same class p^{hit} (nearest hit) and the nearest neighbor prototype of the different class p^{miss} (nearest miss). The algorithm updates each feature weight f_i according to the following criterion:

$$f_i^m = f_i^{m-1} - \frac{diff(x_i, p_i^{hit})}{m} + \sum_{c \neq class(x)} \frac{p(c) diff(x_i, p_i^{miss})}{m} \tag{19}$$

where $p(c)$ is the prior probability of class c, and $diff(,)$ is the distance between the sample and the prototype for the feature i. This algorithm was chosen because of its widespread use and good performance in general feature selection problems. As a result, the higher weight, the more relevant is a feature.

The second technique is related to divergence measures between classes. One of the best-known distance measures utilized for feature selection in multi-class problems is the average Jeffries-Matusita (JM) distance [8]:

$$JM = \sum_{h=1}^{c} \sum_{k>h}^{c} P_h P_k JM_{hk} \tag{20}$$

where

$$JM_{hk} = \sqrt{2(1 - e^{-b_{hk}})}$$

and

$$b_{hk} = \frac{1}{8}(m_h - m_k)^t \left(\frac{S_W^h + S_W^k}{2} \right)^{-1} (m_h - m_k) + \frac{1}{2} \log \left(\frac{\left| \frac{S_W^h + S_W^k}{2} \right|}{\sqrt{|S_W^h||S_W^k|}} \right)$$

P_i is the priori probability of the i-th class, b_{hk} is the Bhattacharyya distance between the classes h and k. S_W^i and m_i are the covariance matrix and the mean vector of the class i, respectively.

In terms of class separability, the higher is the JM distance between two classes, the more separability between them. To obtain suboptimal subsets of features, we have applied a search strategy based on a Sequential Forward Selection applying this distance ((SFS) JM distance). This technique starts from an empty feature subset and adding one feature at a time, reaching a feature subset with the desired cardinality.

4.2 Experiments including background pixels

During the image labelling process, there is always pixels in an image that are not assigned to any class of interest, mainly because they are pixels that either do not clearly belong to some of the predefined classes or they are assigned to a complementary class. The pixels that have not been assigned to any class are labelled as "background" class. In this subsection, we include the background information in the databases for its evaluation.

The experimental results shown in this section about the classification rates correspond to the average classification accuracy obtained by the NN classifier over the five random partitions described previously. The samples in each partition were randomly assigned to the training and test set with equal sizes as follows: VIS = 43902 pixels, NIR = 44758 pixels, HyMap = 37520 pixels, 92AV3C = 2102 pixels.

On the other hand, given the huge size of the data sets and the trouble in computational cost to apply the supervised approaches, particularly in the case of VIS, NIR and HyMap, the following independent partitions with respect to the data sets were randomly extracted maintaining the prior probability of the classes: VIS = 87805 pixels, NIR = 89516 pixels, HyMap = 93804 pixels and 92AV3C = 10512 pixels. Using these databases, the supervised approaches and the proposed DA method were applied in order to obtain a ranking of relevance of the features, that is, of bands.

Fig 3 represents the classification rate with respect to the subset of N bands selected by each method. Note that the proposed DA method obtained better performance with respect to the rest of methods in the case of database of VIS, and similar accuracy for the other three databases (NIR, HyMap and 92AV3C). It is worthwhile mentioning that the DA approach has a good behavior in all cases when choosing the smaller sets of bands (first one to ten), where the decision is more critical.

$ReliefF$ performs poorer with respect to the other approaches except with HyMap image, where the performance of (SFS) JM distance is worse. $ReliefF$ obtains a ranking of relevance for each single feature and the computational cost grows exponentially with respect to the number of samples in the data set.

On the other hand, (SFS) JM distance provides a high classification accuracy, but the computational cost grows exponentially with respect to the number of dimensions. Table 1 shows the computational time in minutes for the tested methods.

In the case of DA, the principal problem arises when we build the *transinformation matrix*. Thus, the different co-occurrences of pixels in each pair of image bands are calculated [7], which represents an important cost in time. On the other hand, when the matrix is built, the proposed DA method obtain the selected features very quickly.

Therefore, for the band selection problem, where there exists high correlation among different features (image bands), the principle of looking for non correlated bands from the different regions of the spectrum, by reducing the mutual information in the ensemble of

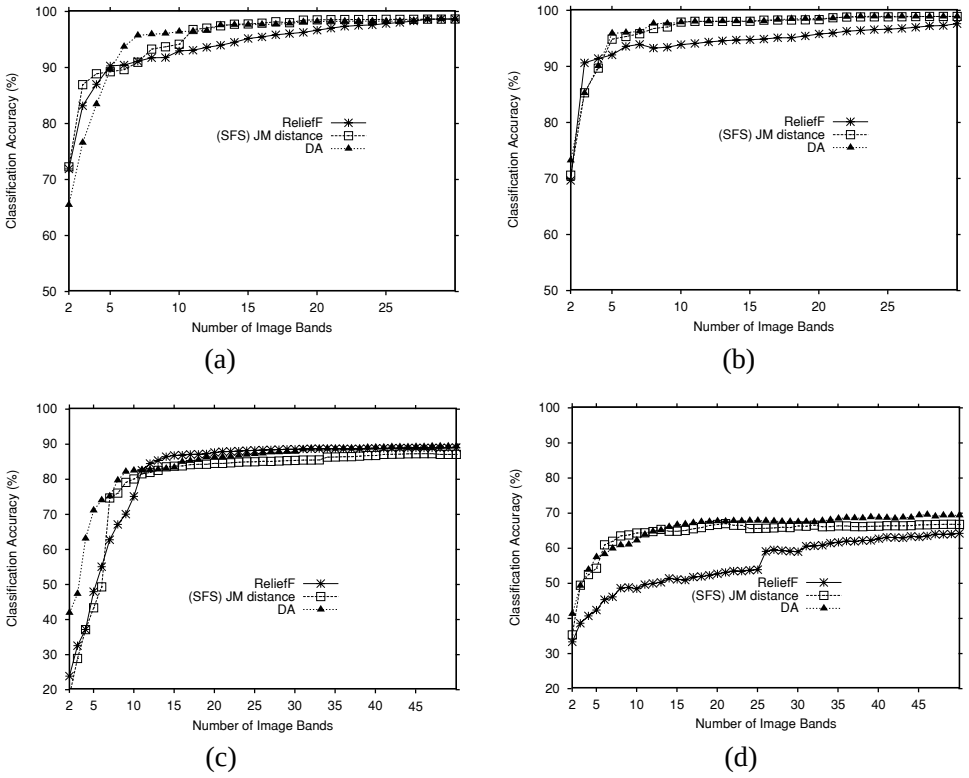


Figure 3: (a) Results over oranges in VIS. (b) Results over oranges in NIR. (c) Results over spectral image with HyMap spectrometer. (d) Results over 92AV3C spectral image. In all cases, it is shown the performance of the *NN* classifier with respect to the number of features obtained by DA, (SFS) JM distance and $ReliefF$.

Table 1: Computational cost in minutes (m) when selecting all features except for (SFS) JM distance, where it is shown for 30 features (VIS and NIR) and 50 features (HyMap and 92AV3C)

Criteria Time (m)				
	VIS	NIR	HyMap	92AV3C
ReliefF	198 m	237 m	423 m	20 m
(SFS)JM distance	17 m	49 m	152 m	151 m
DA (build the matrix)	4 m	8 m	130 m	102 m

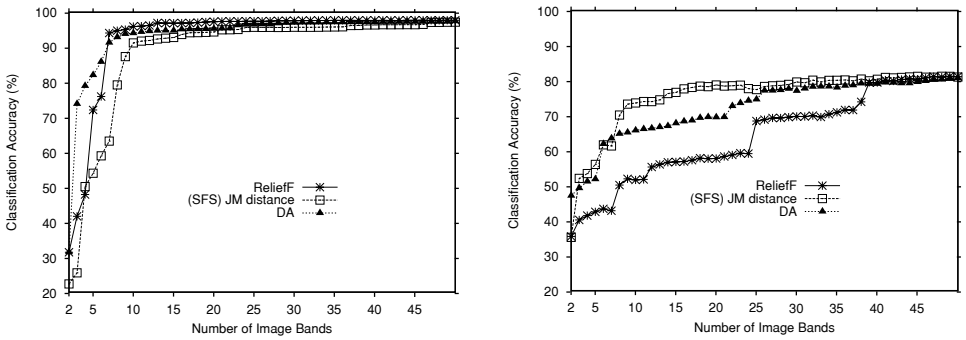


Figure 4: (a) Results over spectral image with HyMap spectrometer. (b) Results over 92AV3C spectral image. In all cases, we show the performance of the *NN* classifier with respect to the number of features obtained by DA, (SFS) JM distance and $ReliefF$.

image bands, has proven to be an effective approach to obtain subsets of selected image bands that also provide satisfactory results from the classification accuracy point of view.

4.3 Experiments without background pixels

The hyperspectral data assigned to the "background class" are usually very scattered and overlapped with other classes, and this fact damages the classification accuracy. Moreover, the elimination of this information supposes a supervised knowledge to detect those regions of the image.

These regions are very difficult to detect with precision from unsupervised information. Therefore, the goal of this experiment is analyzing the advantages that suppose the knowledge of the class distribution without the noise that the *background* class can introduce. In this case, we will focus on HyMap and 92AV3C hyperspectral data, where the background information is much more undefined.

In the case of HyMap, we added the *background* class to the training set and validation set: training = 26190 pixels and validation = 65479 pixels. The test set contains all classes except the *background* class. The total number of test samples is 327336 pixels. Thus, the experiment classifies the test using the ranking of relevance of the features obtained by the validation set with the proposed method and the supervised methods used in the comparison.

The image 92AV3C only contains 10366 instances without the *background* class. Therefore, we apply a holdout partition, where the training and the validation set have the same size with 5181 pixels and the rest of pixels represent the test set = 5185 pixels.

In Fig 3 (a), the best performance is obtained by $Relief$ over HyMap, although DA

reaches a good performance, even better in the first two to seven bands, where the decision is more critical. (SFS) JM distance provides the worst accuracy of the three methods. On the other hand, $Relief$ needs 13 features to reach 96.94 % , while similar experiments realized by Camps et. al. [3] using Support Vector Machines (SVM) only needs 2 features reaching 96.44 %. In this sense, the NN classifier degrades more rapidly than SVM, when the dimension of the input space is lower.

In the case of the image 92AV3C, the NN classifier achieves the best performance using the ranking obtained by (SFS) JM distance. In this case, it exits a clear improving for this method with respect to the other ones. Therefore, the knowledge of the spatial distributions of the sixteen classes allows a better search to pick up goods subset of features.

5 Conclusions and future research

In this work, correlation among image bands in multispectral images has been established in terms of mutual information. The relationships between bands can be represented by the *transinformation matrix*. Using this representation, an approach to rank reduction of the *transinformation matrix* using Deterministic Annealing has been proposed to look for a given number of bands as less correlated as possible among them.

Although the proposed method has not been established in terms of class separability for supervised training sets, it has been shown in the experimental results that the image bands selected by DA provide very satisfactory results with respect to classification accuracy when using the selected bands. This effect is more noticeable when choosing small sets of features, when the decision is more critical. These two advantages, its unsupervised nature and the ability to choose highly relevant bands in the case of small sets, represent the more relevant characteristics of the proposed approach.

Acknowledgments

This has work been supported in part by grants IST-2001-37306 (IST Project European Union) and P1-1B2004-08 from Fundació Caixa Castelló-Bancaixa.

References

[1] J. Aczel, J., Daroczy, Z.: On measures of information and their characterization. New York: Academic Press, 1975.

[2] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional for data mining applications. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, WA, June (1998), 94–105

[3] Camps-Valls, G., Gómez-Chova, L., Calpe-Maravilla, J., Soria-Olivas, E., Martín-Guerrero, J.D., Moreno J.: Support Vector Machines for Crop Classification Using Hyperspectral Data. In 1st. Iberian Conference on Pattern Recognition and Image Analysis, Mallorca, Spain, (2003) 134-141

[4] Hansen, P.C., and Jensen, S.H.: FIR Filter Representation of Reduced-Rank noise Reduction. IEEE Transaction On Signal Processing, **46** (1998) 1737–1741

[5] Groves, P., Bajcsy, P.: Methodology for hyperspectral band and classification model selection. IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data. An Honorary Workshop for Prof. David A. Landgrebe, Washington D.C., 2003.

[6] Jaynes, E.T.: Prior Probatilities. IEEE Transations on System Science and Cybernetic, SSC-4, (1968) pp. 227–241. Reprinted in Concepts and Applications of Modern Decision Models, V.M. Rao Tummala and R. C. Henshaw, eds., (Michigan State University Business Studies Series, 1976).

[7] Sotoca, J.M., Pla F., Klaren A.C.: Unsupervised band selection for multispectral images using information theory. In 17th. International Conference on Pattern Recognition, Cambridge (UK),**3**, (2004) 510–513

[8] Bruzzonne, L., Roli, F., Serpico S.B.: An extension to multiclass cases of the Jeffreys-Matusita distance. IEEE Transactions on Geoscience and Remote Sensing, **33** (1995) 1318–1321

[9] Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. In Proceedings of 7th European Conference on Machine Learning, Catania, Italy,(1994) 171–182

[10] Kumar, S., Ghosh, J., Crawford, M.M.: Best basis feature extraction algorithms for classification of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing, **39**, no. 7, (2001) 1368–1379

[11] Jimenez, L., Landgrebe, D.: Supervised classification in high dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data. IEEE Transactions on System, Man, and Cybernetics, **28**, Part C., no. 1, (1998) 39–54

Problem difficulty analysis for enhanced application of editing and condensing

J.M. Sotoca, R.A. Mollineda, J.S. Sánchez
Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, E-12071 Castelló de la Plana (Spain)
E-mail: {sotoca,mollined,sanchez}@uji.es

Abstract

The Nearest Neighbor classifier constitutes one of the most popular supervised classification methods. It is very simple, intuitive and accurate in a great variety of real-world applications. Despite its simplicity and effectiveness, practical use of this rule has been historically limited due to its high storage requirements and the computational costs involved, as well as the presence of outliers. In order to overcome these drawbacks, it is possible to employ a suitable prototype selection scheme, as a way of storage and computing time reduction and it usually provides some increase in classification accuracy. Nevertheless, in some practical cases prototype selection may even produce a degradation of the classifier effectiveness. From an empirical point of view, it is still difficult to know a priori when this method will provide an appropriate behavior. The present paper tries to predict how appropriate a prototype selection algorithm will result when applied to a particular problem, by characterizing data with a set of complexity measures.

1 Introduction

One of the most widely studied non-parametric classification approaches corresponds to the k-Nearest Neighbor (k-NN) decision rule [3]. Given a set of n previously labeled instances (training set, TS), the k-NN classifier consists of assigning an input sample to the class most frequently represented among the k closest instances in the TS, according to a certain dissimilarity measure. A particular case of this rule is when $k = 1$, in which an input sample is assigned to the class indicated by its closest neighbor.

The asymptotic classification error of the k-NN rule (i.e., when n grows to infinity) tends to the optimal Bayes error rate as $k \to \infty$ and $k/n \to 0$. Moreover, if $k = 1$, the error is bounded by approximately twice the Bayes error [3]. The optimal behavior of this rule in asymptotic classification performance along with a conceptual and implementational simplicity make it a powerful classification technique capable of dealing with arbitrarily complex problems, provided that there is a large enough TS available.

Edited by F.Pla, P.Radeva, J.Vitrià, 2006.

Nevertheless, this theoretical requirement of large TS size is also the main problem using the 1-NN rule because of the seeming necessity of a lot of memory and computational resources. This is why numerous investigations have been concerned with finding new approaches that are efficient with computations. Within this context, many fast algorithms to search for the NN have been proposed. Alternatively, some prototype selection techniques [1, 4, 6] have been directed to reduce the TS size by selecting only the most relevant instances among all the available ones, or by generating new prototypes in locations accurately defined.

On the other hand, in many practical situations the theoretical accuracy can hardly be achieved because of certain inherent weaknesses that significantly reduce the effective applicability of k-NN classifiers in real-world domains. For example, the performance of these rules, as with any non-parametric classification approach, is extremely sensitive to data complexity. In particular, class-overlapping, class-density, and incorrectness or imperfections in the TS can affect the behavior of these classifiers. Other prototype selection methods [5, 10, 13, 14] have been devoted to improve the 1-NN classification performance by eliminating outliers (i.e., noisy, atypical and mislabeled instances) from the original TS, and by reducing the possible overlapping between regions from different classes.

Despite the apparent benefits of most prototype selection algorithms, in some domains these techniques might not achieve the expected results due to certain data characteristics. For this reason, it seems interesting to know a priori the conditions under which the application of a prototype selection scheme can become appropriate. A set of data complexity measures [7, 8] are used in this paper to predict when a prototype selection technique leads to an improvement with respect to the plain 1-NN rule.

2 Some problem difficulty measures

The behavior of classifiers is strongly dependent on data complexity. Usual theoretical analysis consists of searching accuracy bounds, most of them supported by impractical conditions. Meanwhile, empirical analysis is commonly based on weak comparisons of classifier accuracies on a small number of unexplored data sets. Such studies usually ignore the particular geometrical descriptions of class distributions to explain classification results. Various recent papers [7, 8] have introduced the use of measures to characterize the problem difficulty (or data complexity) and to relate such descriptions to classifier performance.

In [7, 8], authors define some problem difficulty measures for two classes. For our purposes, a generalization of such measures for the n-class problem is accomplished. The ideal goal is to represent classification problems as points in a space defined by a number of measures, where clusters can be related to classification performances. Next paragraphs describe the measures selected for the present study (the same short notation as in the original

paper [7] is here used).

2.1 Generalized Fisher's discriminant ratio (F1)

The plain version of this well-known measure computes how separated are two classes according to a specific feature. It compares the difference between class means with the sum of class variances. A possible generalization for C classes, which also considers all feature dimensions, can be stated as follows:

$$F1 = \frac{\sum_{i=1}^{C} n_i \cdot \delta(m, m_i)}{\sum_{i=1}^{C} \sum_{j=1}^{n_i} \delta(x_j^i, m_i)} \tag{1}$$

where n_i denotes the number of samples in class i, δ is a metric, m is the overall mean, m_i is the mean of class i, and x_j^i represents the sample j belonging to class i.

2.2 Volume of overlap region (F2)

The original measure computes, for each feature, the length of the overlap range normalized by the length of the total range in which all values of both classes are distributed. The volume of the overlap region for two classes is the product of normalized lengths of overlapping ranges for all features. Our generalization sums this measure for all pairs of classes, that is,

$$F2 = \sum_{(c_i,c_j)} \prod_{k} \frac{\min\{\max(f_k, c_i), \max(f_k, c_j)\} - \max\{\min(f_k, c_i), \min(f_k, c_j)\}}{\max\{\max(f_k, c_i), \max(f_k, c_j)\} - \min\{\min(f_k, c_i), \min(f_k, c_j)\}} \tag{2}$$

where (c_i, c_j) goes through all pair of classes, k takes feature index values, while $\min(f_k, c_i)$ and $\max(f_k, c_i)$ compute the minimum and maximum values of feature f_k in class c_i, respectively.

2.3 Feature efficiency (F3)

In [7], the feature efficiency is defined as the fraction of points that can be separated by a particular feature. For a two-class problem, the original measure takes the maximum feature efficiency. This paper considers the points in the overlap range (instead of those separated points as in the original formulation). The measure value for C classes is the overall fraction of points in some overlap range of any feature for any pair of classes. Obviously, points in more than one range are counted once. This measure does not take into account the joint contribution of features.

2.4 Non-parametric separability of classes (N2, N3)

The first measure (N2) is the ratio of the average distance to intraclass nearest neighbor and the average distance to interclass nearest neighbor. It compares the intraclass dispersion with the interclass separability. Smaller values suggest more discriminant data. The second measure (N3) is simply the estimated error rate of the 1-NN rule by the leaving-one-out scheme.

2.5 Density measure (T2)

This measure does not characterize the overlapping level, but contributes to understand the behavior of some classification problems. It describes the density of spatial distributions of samples by computing the average number of instances per dimension.

3 Editing and condensing

Prototype Selection techniques have been proposed as a way of minimizing some problems related to the k-NN classifier. They consist of selecting an appropriate reduced subset of instances and applying the 1-NN rule using only the selected examples. Two different families of prototype selection methods exist in the literature: editing and condensing algorithms.

Editing [5, 10, 13–15] eliminates erroneous cases from the original set and "cleans" possible overlapping between regions from different classes, what usually leads to significant improvements in performance. Thus the focus of editing is not on reducing the set size, but on defining a high quality TS by removing outliers. Nevertheless, as a by-product these algorithms also obtain some decrease in size and consequently, a reduction of the computational burden of the 1-NN classifier.

Wilson [14] introduced the first editing proposal. Briefly, this consists of using the k-NN rule to estimate the class of each instance in the TS, and removing those whose class label does not agree with that of the majority of its k neighbors. Note that this algorithm tries to eliminate mislabeled instances from the TS as well as close border instances, smoothing the decision boundaries.

On the other hand, condensing [1, 4, 6, 9, 11, 12] aims at selecting a sufficiently small set of training instances that produces approximately the same performance than the 1-NN rule using the whole TS. It is to be noted that many condensing schemes make sense only when the classes are clustered and well-separated, which constitutes the focus of the editing algorithms.

Hart's algorithm [6] is the earliest attempt at minimizing the number of stored instances by retaining only a *consistent* subset of the original TS. A consistent subset, say S, of a

set of instances, T, is some subset that correctly classifies every instance in T using the 1-NN rule. Although there are usually many consistent subsets, one generally is interested in the *minimal* consistent subset (i.e., the subset with the minimum number of instances) to minimize the cost of storage and computing time. Unfortunately, Hart's algorithm cannot guarantee that the resulting subset is minimal in size.

4 Experiments and results

As already stated in Sect. 1, in some cases prototype selection algorithms may produce an effect different from the one theoretically expected, that is, they may even degrade the performance of the plain 1-NN classifier. A way of characterizing the problems could be by using the data complexity measures introduced in Section 2. Thus the experiments reported in this paper aim at describing the databases in terms of such measures and analyzing the conditions under which prototype selection methods can perform better than the plain 1-NN rule.

In our experiments, we have included 17 data sets taken from the UCI Database Repository (`http://www.ics.uci.edu/~mlearn`) and also from the ELENA European Project (`http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/`). The 5-fold cross-validation error estimate method has been employed for each database: 80% of the available instances have been used as the TS and the rest of instances for the test set. The main characteristics of these data sets are summarized in Table 1. Their values for the complexity measures previously described are summarized in Table 2.

For the prototype selection methods, we have tested Wilson's editing, Hart's condensing, and the *combining* edited and condensed set. In this latter case, we have firstly applied Wilson's editing to the original TS in order to remove mislabeled instances and smooth the decision boundaries, and then Hart's algorithm has been used over the Wilson's edited set to further reduce the number of training examples. After preprocessing the TS by means of some prototype selection scheme, the 1-NN classifier has been applied to the test set.

Table 3 reports the error rate and the percentage of original training instances retained by each method for each database. Typical settings for Wilson's editing algorithm (i.e., number of neighbors) have been tried and the ones leading to the best performance have been finally included. The databases are sorted by the value of F1. By means of the data complexity measures, we have tried different orderings which could give us an indication of the relation between the complexity of a data set and the particular method applied to it. From all those measures, it seems that F1 is the one that better discriminates between the cases in which an editing has to be firstly applied and those in which one could directly employ the plain 1-NN rule.

As can be seen in Table 3, Wilson's editing outperforms the 1-NN rule when F1 is under

Table 1: Some characteristics of the experimental data sets

	Classes	Dim	Samples
Cancer	2	9	683
Clouds	2	2	5000
Diabetes	2	8	768
Gauss	2	2	5000
German	2	24	1000
Glass	6	9	214
Heart	2	13	270
Liver	2	6	345
Phoneme	2	5	5404
Satimage	6	36	6435
Segment	7	19	2310
Sonar	2	60	208
Texture	11	40	5500
Vehicle	4	18	846
Vowel	11	10	528
Wform	3	21	4999
Wine	3	13	178

0.410 (that is, when regions from different classes are strongly overlapped). Consequently, for a particular problem, one could decide to apply an editing to the original TS or directly to employ the plain 1-NN classifier according to the value of F1. For data sets with no (or weak) overlapping (in Table 3, those with F1 > 0.410), the use of an editing can become even harmful in terms of error rate: it seems that editing removes some instances that are defining the decision boundary and therefore, this produces a certain change in the form of such a boundary. Another important result in Table 3 refers to the percentage of training instances given by Hart's condensing: in general, the reductions in TS size for databases with high overlap are lower than those in the case of data sets with weak overlapping.

From the results included in Table 3, it is possible to distinguish between two situations. First, for domains in which the classes are strongly overlapped, one has to employ an editing algorithm in order to obtain a lower error rate (in these cases, benefits in size reduction and classification time are also obtained). Second, for databases with weak overlapping (i.e.,

Table 2: Problem difficulty measures for the experimental data sets

	F1	F2	F3	N2	N3	T2
Cancer	1.315	0.319	0.902	0.220	0.950	76
Clouds	0.245	0.380	0.877	0.019	0.846	2500
Diabetes	0.032	0.252	0.994	0.839	0.679	96
Gauss	0.000	0.309	0.960	0.060	0.650	2500
German	0.026	0.664	0.992	0.794	0.664	42
Glass	0.474	0.013	0.963	0.452	0.734	24
Heart	0.041	0.196	0.985	0.838	0.567	21
Liver	0.017	0.073	0.968	0.853	0.623	58
Phoneme	0.082	0.271	0.878	0.067	0.912	1081
Satimage	2.060	0.000	0.883	0.215	0.909	179
Segment	0.938	0.000	0.583	0.072	0.967	122
Sonar	0.029	0.000	0.947	0.544	0.827	3
Texture	3.614	0.000	0.726	0.119	0.992	138
Vehicle	0.259	0.169	0.968	0.273	0.653	47
Vowel	0.536	0.482	0.962	0.129	0.991	53
Wform	0.410	0.007	0.997	0.769	0.780	238
Wine	2.362	0.000	0.315	0.018	0.770	14

F1 is high enough), in which error rate given by the 1-NN rule can be even lower than that achieved with an editing, one should still decide when to apply a prototype selection scheme (reducing time and storage needs) and when to directly use the 1-NN classifier without any preprocessing. In many problems, differences in error rate are not statistically significant (for example, in Satimage database, the error rates for Wilson's editing and 1-NN rule are 16.90% and 16.40%, respectively) and in such cases, savings in memory requirements and classification times can become the key issues for deciding which method to employ.

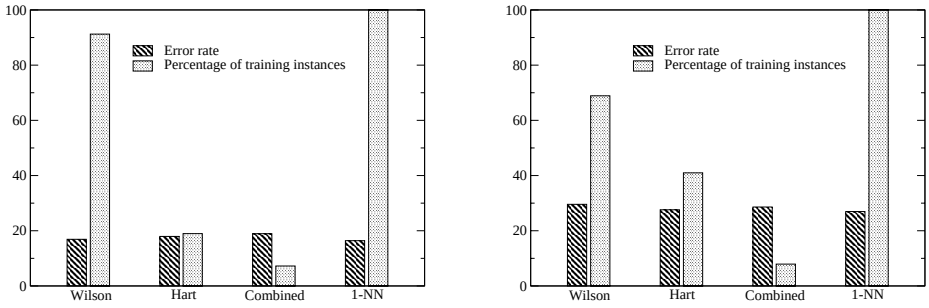
Fig. 1 illustrates the situation just described, comparing the error rate and the percentage of training instances for two databases with a high value of F1. For the Satimage database, differences in error rate are not statistically significant but, in terms of percentage of training instances, the combined approach is clearly the best option: it stores only 7.23% of the original samples and provides an error rate approximately 2% higher than the plain 1-NN rule with the whole TS (100% of instances). Results for the Wine database are similar to

Table 3: 1-NN error rate and percentage of training instances (in brackets), sorted by F1 (values in italics indicate the lowest error rate for each database)

	F1	Wilson		Hart		Combined		1-NN
Gauss	0.000	*30.24*	(68.93)	35.86	(54.07)	30.76	(8.08)	35.06
Liver	0.017	*32.18*	(66.59)	37.68	(59.13)	34.17	(17.46)	34.50
German	0.026	*30.60*	(68.10)	38.50	(53.45)	30.49	(10.73)	34.69
Sonar	0.029	43.03	(82.04)	50.40	(34.49)	*40.42*	(17.25)	47.89
Diabetes	0.032	*27.21*	(71.66)	35.29	(51.47)	27.34	(10.78)	32.68
Heart	0.041	*32.61*	(58.06)	42.14	(59.54)	35.20	(13.52)	41.83
Phoneme	0.082	*26.43*	(89.42)	34.07	(21.55)	28.17	(9.28)	29.74
Clouds	0.245	*11.52*	(88.06)	17.28	(27.25)	11.80	(4.07)	15.34
Vehicle	0.259	36.54	(64.15)	36.76	(53.43)	37.36	(18.65)	*35.59*
Wform	0.410	*18.96*	(82.01)	26.01	(38.96)	21.84	(17.09)	22.04
Glass	0.474	32.37	(70.69)	31.35	(47.01)	32.74	(18.74)	*28.60*
Vowel	0.536	5.23	(96.69)	4.57	(23.40)	8.51	(21.96)	*2.10*
Segment	0.938	5.28	(96.09)	5.88	(13.73)	6.88	(9.90)	*3.72*
Cancer	1.315	*4.25*	(95.54)	6.43	(11.44)	4.39	(3.00)	4.54
Satimage	2.060	16.90	(91.24)	17.94	(18.96)	18.93	(7.23)	*16.40*
Wine	2.362	29.57	(68.89)	27.59	(40.97)	28.60	(7.92)	*26.95*
Texture	3.614	1.22	(98.97)	2.91	(8.01)	2.86	(6.86)	*1.04*

those of the Satimage domain, although now differences in error rate are more important when comparing Wilson's editing and 1-NN classifier. As a conclusion, for these cases with high F1, one has to decide whether it is more important to achieve the lowest error rate but without any reduction in storage or to attain a moderate error rate with important savings in memory requirements (and also, in classification times).

Despite F1 results in the complexity measure with the highest discrimination power in the specific framework of prototype selection, it is to be noted that other measures can become especially useful for other different tasks. For example, F2 and F3 (conveniently adapted) could be particularly interesting in the case of feature selection because they could be used as objective functions to pick subsets of relevant features. On the hand, other measures constitute a complement in the analysis of certain problems. In this sense, T2 can help to understand why the plain 1-NN classifier does not perform well in problems with



(a) Satimage (b) Wine

Figure 1: Comparing error rate and percentage of the original instances retained by each method for several databases with high F1

weak overlapping. For example, the 1-NN error rate in Wine database, which corresponds to a problem with almost no overlapping (F1 = 2.362), is high enough (26.95%); this can be explained by the fact that there exists a very small number of training instances per dimension (T2 = 14).

5 Conclusions and future research

The primary goal of this paper has been to analyze the relation between data complexity and efficiency for the 1-NN classification. More specifically, we have investigated on the utility of a set of complexity measures as a tool to predict whether or not the application of some prototype selection algorithm results appropriate in a particular problem.

After testing different data complexity measures, from the experiments carried out over 17 databases, it seems that F1 can become especially useful to distinguish between the situations in which a prototype selection technique is clearly needed and those in which a more extensive study has to be considered. While in the former case the prototype selection approach achieves the lowest error rate and some savings in memory storage, for the later it is not clear the significance of gains in error rate and therefore, other measures should be employed because even the application of a method with a higher error rate could be justified according to other benefits in computational requirements.

It is worth noting that for those situations in which prototype selection degrades the 1-NN accuracy, one could still reduce the (high) computing time associated to the plain 1-NN rule by means of *fast search* algorithms [2]. However, it is known that fast search algorithms can lessen the number of computations during classification but they still maintain the memory requirements.

Future work is mainly addressed to extend the data complexity measures employed in the same framework of the present paper, trying to better characterize the conditions for an appropriate use of prototype selection techniques. A larger number of editing and condensing algorithms, both from selection and abstraction perspectives, has also to be tested in order to understand the relation between data complexity and performance of the 1-NN classifier. Finally, a more exhaustive study will help to categorize the use of several complexity measures for different pattern recognition tasks.

Acknowledgments

This has work been supported in part by grants TIC2003-08496 from the Spanish CICYT, GV04A/705 from Generalitat Valenciana, and P1-1B2004-08 from Fundació Caixa Castelló-Bancaixa.

References

[1] Chang, C.-L.: Finding prototypes for nearest neighbor classifiers, IEEE Trans. on Computers 23 (1974) 1179-1184.

[2] Chavez, E., Navarro, G., Baeza-Yates, R.A., Marroquin, J.L.: Searching in metric spaces, ACM Computing Surveys 33 (2001) 273-321.

[3] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification, IEEE Trans. on Information Theory 13 (1967) 21-27.

[4] Dasarathy, B.V.: Minimal consistent subset (MCS) identification for optimal nearest neighbor decision systems design, IEEE Trans. on Systems, Man, and Cybernetics 24 (1994) 511-517.

[5] Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, NJ (1982).

[6] Hart, P.E.: The condensed nearest neighbor rule, IEEE Trans. on Information Theory 14 (1968) 515-516.

[7] Ho, T.-K., Basu, M.: Complexity measures of supervised classification problems, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (2002) 289-300.

[8] Bernardo, E., Ho, T.-K.: On classifier domain of competence, In: Proc. 17th. Int. Conf. on Pattern Recognition 1, Cambridge, UK (2004) 136-139.

[9] Kim, S.-W., Oommen, B.J.: Enhancing prototype reduction schemes with LVQ3-type algorithms, Pattern Recognition 36 (2003) 1083-1093.

[10] Kuncheva, L.I.: Editing for the k-nearest neighbors rule by a genetic algorithm, Pattern Recognition Letters 16 (1995) 809-814.

[11] Mollineda, R.A., Ferri, F.J., Vidal, E.: An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering, Pattern Recognition 35 (2002) 2771-2782.

[12] Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An algorithm for a selective nearest neighbour decision rule, IEEE Trans. on Information Theory 21 (1975) 665-669.

[13] Tomek, I.: An experiment with the edited nearest neighbor rule, IEEE Trans. on Systems, Man and Cybernetics 6 (1976) 448-452.

[14] Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets, IEEE Trans. on Systems, Man and Cybernetics 2 (1972) 408-421.

[15] Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms, Machine Learning 38 (2000) 257-286.

Comparison of dynamic and static weighting functions for classifier fusion

R. M. Valdovinos[1], J. S. Sánchez[2]
[1]Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, México
li_rmvr@hotmail.com
[2] Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló, Spain
sanchez@uji.es

Abstract

The simple majority voting scheme constitutes one of the most popular techniques to perform the classifier fusion in an ensemble of classifiers. However, when the performance of the ensemble members is not uniform, the efficiency of this type of voting results affected negatively. In this paper, an experimental comparison between simple and weighted voting (both dynamic and static) is presented. New weighting methods in the direction of the dynamic approach are also introduced. Experimental results with several real-problem data sets demonstrate the advantages of the weighting strategies over the simple voting scheme. When comparing the dynamic and the static approaches, results show that the dynamic weighting is generally superior to the static strategy in terms of classification accuracy.

Keywords: Multiple classifier systems; fusion; voting; weighting; nearest neighbor.

1 Introduction

A multiple classifier system (MCS) is a set of individual classifiers whose decisions are combined when classifying new patterns. There are many different reasons for combining multiple classifiers to solve a given learning problem [6], [12]. First, MCSs try to exploit the local different behavior of the individual classifiers to improve the accuracy of the overall system. Second, in some cases MCS might not be better than the single best classifier but can diminish or eliminate the risk of picking an inadequate single classifier. Another reason for using MCS arises from the limited representational capability of learning algorithms. It is possible that the classifier space considered for the problem does not contain the optimal classifier.

Let D = { D_1, ..., D_h } be a set of classifiers. Each classifier assigns an input feature vector $\mathbf{x} \in \Re^n$ to one of the c problem classes. The output of a MCS is an h-dimensional vector containing the decisions of each of the h individual classifiers:

$$[D_1(\mathbf{x}),..., D_h(\mathbf{x})]^T \quad (\mathbf{1})$$

It is accepted that there are two main strategies in combining classifiers: selection and fusion. In classifier selection, each individual classifier is supposed to be an expert in a part of the feature space and therefore, we select only one classifier to label the input vector **x**. In classifier fusion, each component is supposed to have knowledge of the whole feature space and correspondingly, all individual classifiers decide the label of the input vector.

Focusing on the fusion strategy, the combination can be made in many different ways. The simplest one employs the majority rule in a plain voting system [4]. More elaborated schemes use weighted voting rules, in which each individual component is associated with a different weight [5]. The final decision can be made by majority, average [6], minority, medium [7], product of votes, or using some other more complex methods [8], [9], [10], [19].

In the present work, some methods for weighting the individual components in a MCS are proposed, and their effectiveness is empirically tested over real data sets. Three of these methods correspond to the so-called dynamic weighting, by using the distances to a pattern. The last method, which belongs to the static weighting strategy, estimates the leaving-one-out error produced by each classifier in order to set the weights of each component [21].

From now on, the rest of the paper is organized as follows. Section 2 provides a brief review of the main issues related to classifier fusion and makes a very simple categorization of weighting methods, distinguishing between dynamic and static weighting of classifiers. In Section 3, seveal weighting procedures are also introduced. The experimental results are discussed in Section 4. Finally, some conclusions and possible further extensions are given in Section 5.

2 Classifier fusion and voting schemes

As pointed out in the previous section, classifier fusion assumes that all individual classifiers are competitive, instead of complementary. For this reason, each component takes part in the decision of classifying an input test pattern.

In the simple voting (by majority), the final decision is taken according to the number of votes given by the individual classifiers to each one of the classes, thus assigning the

test pattern to the class that has obtained a majority of votes. When working with data sets that contain more than two classes, in the final decision ties among some classes are very frequently obtained. To solve this problem, several criteria can be considered. For instance, to randomly take the decision, or to implement an additional classifier whose ultimate goal is to bias the decision toward a certain class [15].

An important issue that has strongly called the attention of many researchers is the error rate associated to the simple voting method and to the individual components of a MCS. Hansen and Salomon [17] show that if each one of the classifiers being combined has an error rate less than 50%, it may be expected that the accuracy of the ensemble improve when more components are added to the system. However, this assumption not always is fulfilled. In this context, Matan [18] asserts that in some cases, the simple voting might perform even worse than any of the members of the MCS. Thus some weighting method can be employed in order to partially overcome these difficulties.

A weighted voting method has the potential to make the MCS more robust to the choice of the number of individual classifiers. Two general approaches to weighting can be remarked: dynamic weighting and static weighting of classifiers. In the dynamic strategy, the weights assigned to the individual classifiers can change for each test pattern. On the contrary, in the static weighting, the weights are computed for each classifier in the training phase, and they are maintained constant during the classification of the test patterns.

3 New weighting functions for classifier fusion

In the following sections, several weighting functions, both from the dynamic and the static categories, are explored. It has to be noted that in the present work, all the individual classifiers correspond to the 1-NN (Nearest Neighbor) rule [16]. This is a well-known supervised non-parametric classifier that combines conceptual and implementational simplicity with an asymptotic error rate conveniently bounded in terms of the optimal Bayes error. In its classical manifestation, given a set of m previously labeled instances (or training set, TS), this classifier assigns any input test pattern to the class indicated by the label of the closest example in the TS. The extension of this rule corresponds to the k-NN classifier, which consists of assigning an input pattern to the class most frequently represented among the k closest training instances.

3.1 Dudani's dynamic weighting

A weighted k-NN rule for classifying new patterns was first proposed by Dudani [3]. The votes of the k nearest neighbors are weighted by a function of their distance to the test

pattern. In his original proposal, a neighbor with smaller distance is weighted more heavily than one with a greater distance: the nearest neighbor gets a weight of 1, the furthest neighbor a weight of 0, and the other weights are scaled linearly to the interval in between (Eq. 2):

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where d_j denotes the distance of the j'th nearest neighbor to the test pattern, d_1 is the distance of the nearest neighbor, and d_k indicates the distance of the furthest (k'th) neighbor.

Now, this function will be here applied to make the dynamic weighting of the individual components in an ensemble. Correspondingly, the value of k (that is, the number of nearest neighbors in Dudani's rule) will be replaced by the number of classifiers h that constitute the MCS. The procedure to assign the weights can be described as follows:

1. Let d_j (j = 1, …, h) be the distance of an input test vector **x** to its nearest neighbor in the j'th individual classifier.
2. Sort the h distances in increasing order: d_1, …, d_h.
3. Weight classifier D_j by means of function in Eq. 2.

3.2 Dynamic weighting by index

Another weighting function is here considered. Like in Dudani's method, the h distances of the test pattern **x** to its nearest neighbor in each individual classifier have also to be sorted. In this case, each classifier D_j is weighted according to the following function:

$$w_j = h - j + 1 \tag{3}$$

where j represents the index of an individual classifier after sorting the corresponding h distances.

Consider a MCS consisting of three individual classifiers D = {D_1, D_2, D_3}. The distance of the nearest neighbor to a given test pattern **x** by means of each classifier is d_1, d_2, and d_3, respectively. Now suppose that $d_2 < d_1 < d_3$. Thus after sorting the three distances, the index of classifier D_1 is 2, the index of D_2 is 1, and the index of D_3 is 3.

Consequently, by applying the weighting function in Eq. 3, the resulting weights are w_1 = 3 – 2 + 1 = 2, w_2 = 3 – 1 + 1 = 3, and w_3 = 3 – 3 +1 = 1.

3.3 Dynamic weighting by averaged distances

We here propose a novel weighting function, which is based on the computation of averaged distances. In summary, the aim of this new dynamic weighting procedure is to reward (by assigning the highest weight) the individual classifier with the nearest neighbor to the input test pattern. The rationale behind this is that such a classifier probably corresponds to that with the highest accuracy in the classification of the given test pattern. Thus each classifier D_j will be weighted by means of the function shown in Eq. 4:

$$w_j = \frac{\sum_{i=1}^{h} d_i}{d_j} \tag{4}$$

Note that, by using this weighting function, we effectively accomplish the goal previously stated, that is, the individual classifier with the smallest distance will get the highest weight, while the one with the greatest distance will obtain the lowest weight.

3.4 Static weighting by leaving-one-out error estimate

While the previous methods weight the individual components of a MCS in a dynamic manner, the last proposal corresponds to the static category. In this sense, weighting will be here performed in the training phase by means of the leaving-one-out error estimate method. To this end, for each individual classifier D*j*, the following function e_j is defined:

$$e_j = \frac{1}{m} \sum_{x \in S} e(y, x) \tag{5}$$

where m denotes the number of patterns in a training sample S, x represents a training instance, y is the nearest neighbor of x in $S - \{x\}$, and $e(y, x)$ is defined as follows:

$$e(y, x) = \begin{cases} 0 & \text{if } L(y) = L(x) \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where $L(x)$ is the class label of a pattern x, and $L(y)$ indicates the class label of a pattern y.

By using the error function just introduced, each individual classifier D_j will be weighted according to the function in Eq. 7:

$$w_j = 1 - \frac{e_j / m}{\sum_{i=1}^{h} e_i} \tag{7}$$

Note that this weight is directly related to the amount of errors produced by each individual classifier. Thus the classifier with the smallest error will be assigned the highest weight, while the one with the greatest error will obtain the lowest weight.

4 Experimental results

The results here reported correspond to the experiments over six real data sets taken from the UCI Machine Learning Database Repository [11]. For each data set, the 5-fold cross-validation error estimate method was employed: 80% of the available patterns were for training purposes and 20% for the test set.

The integration of the MCS was performed by manipulating the patterns [12] for each of the classes, thus obtaining three different individual classifiers with four variants:

- Sequential selection [1], [2] (Sel1)
- Random selection with no replacement [1], [2] (Sel2)
- Selection with Bagging [13] (Sel3)
- Selection with Boosting [14] (Sel4)

The experimental results given in Table 1 correspond to the averages of the general accuracy in the fusion, by technique of pattern selection and method of weighting. The 1-NN classification accuracy for each entire original TS (i.e., with no combination) has also been included as the baseline classifier. Analogously, the results for the MCS with simple voting (no weighting) are reported for comparison purposes.

From results in Table 1, some preliminary conclusions can be drawn. First, for all data sets there exists at least one classifier fusion technique whose classification accuracy is higher than that obtained when using the whole TS (i.e., with no combination). Second, comparing the four selection methods, in general Sel1 and Sel4 clearly outperform the other two selection approaches (namely, random with no replacement and bagging), independent of the voting scheme adopted. On the other hand, focusing on sequential

selection (Sel1) and boosting (Sel4), the accuracy of Sel1 results superior to that of Sel4 in most cases (22 out of 30).

	Cancer	Heart	Liver	Pima	Glass	Vehicle
Original TS	95.62	58.15	65.22	65.88	70.00	64.24
Simple voting						
Sel1	*96.93*	*65.19*	*63.77*	68.89	*68.00*	*64.48*
Sel2	66.42	50.37	57.10	59.35	56.50	62.10
Sel3	72.12	45.19	50.14	60.00	60.50	60.55
Sel4	94.16	57.78	62.03	*70.07*	62.50	60.43
Dudani's weighting						
Sel1	95.62	58.15	***65.51***	*68.37*	*70.00*	*64.24*
Sel2	68.47	52.96	56.23	59.08	67.00	61.02
Sel3	74.16	47.41	52.17	60.26	65.00	60.91
Sel4	*95.89*	*58.52*	60.87	67.58	66.50	64.24
Weighting by index						
Sel1	95.91	*61.11*	*62.61*	68.24	***71.00***	*64.48*
Sel2	65.84	54.07	53.04	62.09	62.00	62.34
Sel3	72.41	47.78	49.28	60.92	61.50	60.79
Sel4	***99.27***	57.41	59.42	*70.07*	66.00	62.81
Weighting by averaged distances						
Sel1	*96.50*	***65.56***	*65.22*	68.37	*68.00*	***64.72***
Sel2	62.04	49.63	57.10	59.08	59.00	59.00
Sel3	70.80	45.93	50.14	60.26	62.50	63.41
Sel4	93.58	57.78	62.32	***70.85***	63.00	61.50
Static weighting						
Sel1	*96.93*	*65.19*	*63.77*	68.89	*68.50*	*63.65*
Sel2	66.42	50.37	57.10	59.35	56.00	62.93
Sel3	72.12	45.19	50.14	60.00	60.50	59.84
Sel4	94.16	59.63	62.03	*70.07*	63.00	61.03

Table 1: Averaged accuracy of different classifier fusion methods. Values in italics indicate the best selection method for each voting scheme and each data set. Boldface is used to emphasize the highest accuracy for each problem

If we now compare the simple and the weighted voting schemes, we can observe that in all data sets, we can find a weighting technique with better results than those of the simple majority voting. The Dudani's weighting outperforms all the other methods in

Liver database. The weighting by index is the best in Cancer and Glass domains. The weighting by averaged distances achieves the highest accuracy in Heart, Pima and Vehicle databases.

Finally, with respect to differences in accuracy between dynamic and static weighting, it has to be especially remarked the fact that results of the static strategy are always inferior to those of the dynamic approach. As can be seen, although differences are not significant, the static weighting does not seem to present any advantage with respect to the dynamic weightings.

5 Conclusions and future work

In a MCS, performance mainly depends on the accuracy of the individual classifiers and on the specific way of combining the individual decisions. Correspondingly, it results crucial to appropriately handle the combination of decisions in order to attain the most accurate system. In the present work, several weighting methods, both from the dynamic and static approaches, have been introduced and empirically compared with the simple majority voting scheme.

From the experiments carried out, our study shows that the weighting voting clearly outperforms the simple voting procedure, which erroneously assumes the uniform performance of the individual components of a MCS. Another issue to remark is that the dynamic weighting is superior to the static strategy, in terms of classification accuracy.

At this moment, it has to be admitted that it results difficult enough to propose one of the dynamic weightings as the best method. In fact, differences among them are more or less significant depending on each particular database. Nevertheless, one can see that the weighting by averaged distances achieves the highest accuracy in 3 out of 6 problems (50% of the cases), while the weighting by index in 2 out of 6 databases (33% of the cases).

Future work is primarily addressed to investigate other weighting functions applied to classifier fusion. For instance, the inverse distance function proposed by Shepard [20] could represent a good alternative to other weighted voting schemes with low classification accuracy. On the other hand, the results reported in this paper should be viewed as a first step towards a more complete understanding of the behavior of the weighted voting procedures and consequently, it is still necessary to perform a more extensive analysis of the dynamic and static weighting strategies over a larger number of synthetic and real problems.

Acknowledgements

This work has been partially supported by grant TIC2003-08496 from the Spanish CICYT.

References

1. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers, Pattern Analysis and Applications 6 (2003) 245-256.
2. Valdovinos, R.M., Barandela, R.: Sistema de Múltiples Clasificadores. Una alternativa para la Escalabilidad de Algoritmos, In: Proc. of the 9th Intl. Conference of Research on Computer Sciences, Puebla, Mexico (2002).
3. Dudani, S.A.: The distance weighted k-nearest neighbor rule, IEEE Trans. on Systems, Man and Cybernetics 6 (1976) 325-327.
4. Kuncheva, L.I., Kountchev, R.K.: Generating classifier outputs of fixed accuracy and diversity, Pattern Recognition Letters 23 (2002) 593–600.
5. Woods, K., Kegelmeyer Jr., W.P, Bowyer, K.,: Combination of multiple classifiers using local accuracy estimates, IEEE Trans. on Pattern Analysis and Machine Intelligence 19 (1997) 405-410.
6. Kuncheva, L.I.: Using measures of similarity and inclusion for multiple classifier fusion by decision templates, Fuzzy Sets and Systems 122 (2001) 401-407.
7. Chen, D., Cheng, X.: An asymptotic analysis of some expert fusion methods, Pattern Recognition Letters 22 (2001) 901–904.
8. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion, Pattern Recognition 34 (2001) 299-314.
9. Ho, T.-K.: Complexity of classification problems and comparative advantages of combined classifiers, In: Proc. of the 1st Intl. Workshop on Multiple Classifier Systems, Springer (2000) 97-106.
10. Bahler, D., Navarro, L.: Methods for combining heterogeneous sets of classifiers, In: Proc. of the 17th Natl. Conference on Artificial Intelligence (AAAI-2000), Workshop on New Research Problems for Machine Learning (2000).
11. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, Univ. of California, Irvine, CA (1998).
12. Dietterich, G.T.: Machine learning research: four current directions, AI Magazine 18 (1997) 97–136.
13. Breiman, L.: Bagging predictors, Machine Learning 24 (1996) 123-140.
14. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm, In: Proc. of the 13th Intl. Conference on Machine Learning, Morgan Kaufmann (1996) 148-156.

15. Kubat, M., Cooperson Jr., M.: Voting nearest neighbor subclassifiers, In: Proc. of the 17th Intl. Conference on Machine Learning, Morgan Kaufmann, Stanford, CA (2000) 503-510.
16. Dasaraty, B.V..: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, IEEE Computer Society press, Los Alamitos, CA (1991).
17. Hansen, L.K., Salomon, P. : Neural network ensembles, IEEE Trans. on Pattern Analysis and Machine Intelligence 12 (1990) 993-1001.
18. Matan, O.: On voting ensembles of classifiers, In: Proc. of the 13th Natl. Conference on Artificial Intelligence (AAAI-96), Workshop on Integrating Multiple Learned Models (1996) 84–88.
19. Ho, T.-K., Hull, J.J., Srihari, S.N.: Combination of Decisions by Multiple Classifiers, Structured Document Image Analysis, In: Springer-Verlag, Heidelberg (1992) 188–202.
20. Shepard, R.N.: Toward a universal law of generalization for psychological science, Science 237 (1987) 1317-1323.
21. Verikasa, A., Lipnickasb A., Malmqvista, K., Bacauskieneb, M., Gelzinisb, A.: Soft combination of neural classifiers: a comparative study, Pattern Recognition Letters 20 (1999) 429-444.

Nearest neighbor learning by means of labelled and unlabelled data

F. Vázquez†, J.S. Sánchez‡, F. Pla‡
†Dept. de Computación, Universidad de Oriente
Av. Patricio Lumumba s/n, 90100 Santiago de Cuba, Cuba
E-mail: fvazquez@csd.uo.edu.cu
‡ Dept. de Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain
E-mail: {pla, sanchez}@uji.es

Abstract

A classification system with the capability of continuously increasing its knowledge during the operational phase is here discussed. This idea is strongly related to learning in partially supervised environments in the sense that at the start, the system has only a (possibly) reduced number of labelled instances, but this current knowledge will be progressively increased during the classification of new unlabelled patterns. The learning system proposed in the present paper is based on the popular nearest neighbor classifier and some related techniques. The effectiveness of the algorithm is experimentally evaluated using some benchmark data sets taken from the UCI Machine Learning Database Repository.

1 Introduction

Learning algorithms have been traditionally sorted into two broad categories: supervised and unsupervised, depending on whether labelled data is available or not. In a supervised scenario, the learner is based on the information supplied by a a set of labelled instances (training set, TS) that are assumed to correctly represent all the relevant classes. Violation of this assumption may seriously deteriorate the final classification accuracy.

Supervised classification methods usually operate in two steps: a) the *learning or training phase*, for the system to acquire the necessary knowledge from the labelled instances to make itself able to differentiate among the regarded classes; and b) the *classification or operational phase*, wherein the system proceeds to identify new unknown cases as members of the considered classes. Second stage is not started before completion of the first one and thereafter, no new knowledge is attained.

In the unsupervised learning problem, the learner is provided with only unlabelled examples. The task is to find "clusters" or groups of similar cases that probably correspond to

the underlying classes. Unsupervised learning is often applied to discover structure, regularities or categories in the data, but typically requires human analysis to determine whether the discovered regularities are interesting, and to determine the correspondence between clusters and meaningful categories.

Since the early 90's a third approach to learning, namely *partially supervised*, has received much attention [2–4, 14, 15, 18]. This paradigm conceptually represents a compromise between supervised and unsupervised learning, thus using a (generally) small number of labelled instances together with a (possibly) large set of unlabelled samples. Relevance of partially supervised learning systems is due to the fact that in many practical applications, collecting labelled training instances can be costly and time-consuming, while it is frequently easy to obtain unlabelled examples. Consequently, it results interesting to develop algorithms capable of employing both labelled and unlabelled data for classification. Learning from partially labelled data is also referred to as *semi-supervised learning* [1, 13].

This paper presents an idea to implement a classification system that not only can learn by operating with the labelled training instances, but could also benefit from the experience obtained when classifying new unlabelled patterns. The approach for working with "ongoing learning" presents some advantages: the classifier is more robust because errors or omissions in the original TS can be further corrected during operation, and the system is capable to continue adapting itself to a possibly changing environment.

The ultimate aim is to facilitate the learning system to progressively increase its knowledge and consequently, to enhance the final classification accuracy. In our proposal, the nearest neighbor (NN) rule is employed as the central classifier, mainly because of its flexibility. Because a basic goal is to make the ongoing learning procedure as automatic as possible, it has been designed to work by incorporating new examples into the TS after they have been labelled by the own system. This way, however, presents the danger of performance deterioration by the inclusion of potentially mislabelled patterns to the TS. In order to minimize the risk of introducing these errors, we will employ some filters that detect and discard those mislabelled cases.

From now on, the rest of the paper is organized as follows. Section 2 provides a general description of the k-NN rule along with the most important pros and cons of using this classifier. Section 2 also describes an editing algorithm based on an estimation of probabilities. In Sect. 3, we introduce the ongoing learning system proposed in the present paper. Next, Sect. 4 provides the results obtained from a preliminary empirical study. Finally, the main conclusions and possible directions for future research are outlined in Sect. 5.

2 The k-nearest neighbors classifier

One of the most widely studied supervised classification approaches corresponds to the k-NN decision rule [6]. In brief, given a set of n previously labelled examples, say $X = \{(x_1, \omega_1), (x_2, \omega_2), \ldots, (x_n, \omega_n)\}$, the k-NN classifier consists of assigning an input sample x to the class most frequently represented among the k closest instances in the TS, according to a certain similarity measure (generally, the Euclidean distance metric). A particular case of this rule is when $k = 1$, in which an input sample is decided to belong to the class indicated by its closest neighbor.

Several properties make the k-NN classifier quite attractive, including the fact that the asymptotic risk (i.e., when $n \to \infty$) tends to the optimal Bayes risk as $k \to \infty$ and $k/n \to 0$ [5]. If $k = 1$, the upper bound of the classification error rate is approximately twice the Bayes error [6]. The optimal behavior of this rule in asymptotic classification performance along with a conceptual and implementational simplicity make it a powerful classification technique capable of dealing with arbitrarily complex problems, provided that there is a large enough number of training instances available.

However, in many practical situations, such a theoretical maximum can hardly be achieved due to certain inherent weaknesses that significantly reduce the effective applicability of k-NN classifiers. In particular, the performance of these rules, as with any non-parametric classification approach, is extremely sensitive to data complexity [7].

For example, classification accuracy of k-NN classifiers significantly drops down in domains where many data attributes are irrelevant [16]. Such attributes inappropriately affect the values returned by most dissimilarity metrics. Another problem using the k-NN rule refers to the seeming necessity of a lot of memory and computational resources (especially, in applications with a huge number of training examples). Moreover, these classifiers cannot be straightforwardly employed in domains with missing attributes. Also, the class imbalance (i.e., high differences in class distributions) has been reported as an obstacle on applying distance-based algorithms to real-world problems [11].

On the other hand, class overlapping and noise or imperfections in the TS negatively affect the performance of the k-NN classifiers, and this has been widely demonstrated in many empirical studies (e.g., see [17]). That is the reason why a considerable amount of works have been devoted to improve the classification accuracy by eliminating outliers from the original TS and also cleaning possible overlapping between classes. This strategy has generally been referred to as *editing* [9].

The general idea behind almost any editing procedure consists of estimating the true classification of instances in the TS to retain only those which are correctly labelled. Differences among most editing schemes refer to the classification rule employed for editing purposes along with the error estimate and the stopping criterion [10].

The first proposal to select a representative subset of labelled instances corresponds to Wilson's editing [21], in which a k-NN classifier is used to keep in the TS only "good" examples (that is, training instances that result correctly classified by the k-NN rule). Tomek [19] extended this scheme with a procedure that utilized all the l-NN classifiers, with l ranging from 1 through k, for a given value of k.

A slight modification of the original Wilson's algorithm consists of using, instead of the k-NN classifier, an alternative rule based on the k nearest centroid neighbors (k-NCN) [17], which has been proven to be superior to the traditional k-NN classifier in many practical situations. This kind of neighborhood is defined taking into account not only the proximity of instances to a given input pattern but also their symmetrical distribution around it.

2.1 Estimating class conditional probabilities for editing

Recently, new editing schemes have been proposed, in which the elimination rule is based on an estimation of the probability of each training instance to belong to a certain class, that is, considering the form of the underlying probability distribution in the neighborhood of a point [20]. In order to estimate the values of these distributions, we can compute the distance between a given sample and the training instances.

Given a sample, the closer an instance, the more likely this sample belongs to the same class as the one of such an instance. Accordingly, let us define the probability $P_i(x)$ that a sample x belongs to a class i as:

$$P_i(x) = \sum_{j=1}^{k} p_i^j \frac{1}{1 + \delta(x, x^j)} \tag{1}$$

where p_i^j denotes the probability that the k nearest neighbor x^j belongs to class i, and δ represents a certain distance function. Initially, the values of p_i^j for each instance are set to 1 for its class label assigned in the TS, and 0 otherwise.

The meaning of the above expression states that the probability that a sample x belongs to a class i is the weighted average of the probabilities that its k nearest neighbors belong to that class. The weight is inversely proportional to the distance from the sample to the corresponding k nearest neighbors. From this, we can derive a new decision rule, namely k-Prob, in which a new sample x will be assigned to the class whose probability $P_i(x)$ is maximum.

Following the general scheme of Wilson's editing, the new algorithms consist of eliminating from the TS those instances whose label does not coincide with that assigned by the decision rule based on class conditional probabilities (k-Prob).

A further extension to this proposal consists of considering a threshold, $0 < \mu < 1$,

in the classification rule, with the aim of eliminating those instances whose probability to belong to the class assigned by the rule is not significant. Correspondingly, we are removing samples from the TS that are in the decision borders, where the class conditional probabilities overlap and are confusing, in order to obtain edited sets whose instances have a high probability of belonging to the class assigned in the TS.

3 The use of unlabelled data to increase knowledge

A basic goal of the learning system presented in this paper is to make it as automatic as possible. Accordingly, the procedure has been designed to work by incorporating new patterns into the TS after they have been labelled by the own system (without the participation of a human expert). However, it is evident that this working method can be self-defeating, in the sense that these new training elements would have the class label directly assigned by the decision rule. Therefore, there is the risk to incorporate several mislabelled cases into the TS and consequently, to degrade the overall system accuracy. The system we have designed attempts to overcome such a difficulty by employing some editing algorithms.

On the other hand, albeit the training instances are generally labelled by human experts (or, at least, under their supervision), it is possible to introduce errors into the TS. Thus our initial task will consists of looking for outliers in the TS in order to obtain a collection of correctly labelled instances. In summary, the learning procedure for partially supervised domains consists of the following steps:

1) Initial TS is stored in memory.

2) A first filter is applied to the original TS in order to remove possible noisy instances. As a by-product, it also produces a reduction in the TS size. The resulting edited set will be here referred to as *base knowledge*.

3) Classification phase starts with the base knowledge as the TS.

4) The set of new labelled patterns (those classified during the previous step) is now edited in order to detect possible misclassifications. The patterns identified as erroneous by the editing algorithm will be removed from that set.

5) The base knowledge is now updated by incorporating the new labelled patterns that have not been discarded in the previous step.

6) Return to Step 3 with the new base knowledge.

For the filters considered in this procedure, one could employ any editing algorithm. In the present paper, we have applied two of the schemes introduced in Sect. 2: the k-NCN

editing, and the first algorithm based on class conditional probabilities, namely Wilson-Prob [20]. Analogously, the classification phase (Step 3) can be performed by applying any classifier. Here we have used the classical k-NN rule, the k-NCN classifier, and the new k-Prob decision scheme.

Note that the original base knowledge constitutes the only supervised element of our learning system. The unsupervised component comes from the unlabelled patterns that are sequentially classified and edited by the own system.

Dasarathy [8] proposed a decision system with a design very related to ours. He was also concerned with the robustness of the system through varying domains and with the problem of unrepresentative pre-training. The latter is what he called "partially exposed environments". Consequently, Dasarathy presented an on-line adaptive learning system with two capabilities: a) to progressively improve the classification of patterns belonging to the known classes and, b) to detect the objects not belonging to the currently known classes

However, Dasarathy's system requires the steady participation of a human expert to be in charge of the evaluation of the labels assigned by the system to new patterns and to decide which of them are to be incorporated into the TS. Unfortunately, in real-world operational phase, such operator supervision may be unavailable. We avoid this bottleneck by including in our procedure the necessary tools to allow the system to decide which pieces of new knowledge are trustworthy enough to be accepted.

4 Experimental results

In our experiments, we have included four data sets taken from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/~mlearn`). A number of different partitions were randomly produced for each data set, all keeping the a priori class probabilities. One of these partitions is used as the initial TS, one as an independent validation set, and the rest will be employed as sets of unlabelled data in order to simulate the sequence required for developing the capacity of increasing the knowledge by means of the algorithm presented in the previous section.

Data set	Classes	Features	Size	% Class 1	% Class 2	Partitions
Breast	2	9	683	65.2	34.8	10
Diabetes	2	8	786	34.9	65.1	11
German	2	24	1002	70.4	29.6	14
Heart	2	13	270	55.6	44.4	9

Table 1: A brief summary of the UCI databases used in the experiments.

The main characteristics of the data sets used in the present experiments are summarized in Table 1. The column "Partitions" indicates the total number of random partitions produced for each database. This number means that, for example, in Breast database the classification system will have 8 opportunities to increase its base knowledge, that is, the number of sets with unlabelled data. By this, it is evident that the amount of labelled instances is much smaller than that of the unlabelled patterns. The reason is that, as already stated in Sect. 1, in real applications collecting labelled examples often becomes a costly and difficult process, thus we are here reproducing this practical situation.

The experiments consist of comparing the 1-NN classification accuracy when using the initial TS with that obtained when incorporating the new labelled patterns to the TS after processing each of the partitions. The aim is to evaluate the capacity of increasing the knowledge with the application of our learning algorithm in a partially supervised environment.

t	Alg1	Alg2	Alg3	Alg4
0	92.54	92.54	92.54	92.54
1	94.03	94.03	**94.03**	94.03
2	94.03	94.03	94.03	94.03
3	94.03	94.03	94.03	94.03
4	**95.52**	94.03	92.54	**95.52**
5	95.52	94.03	92.54	95.52
6	95.52	94.03	92.54	95.52
7	95.52	**95.52**	94.03	95.52
8	95.52	95.52	94.03	95.52
1-NN	92.48			

Table 2: Classification accuracies for Breast database (1-NN indicates the classification accuracy when using the original TS without any editing).

Tables 2, 3, 4 and 5 provide the classification accuracies over the different databases used in the present experiments. Column t refers to each partition included in the process. Thus $t = 0$ represents the initial base knowledge, that is, the original TS after being edited. The set obtained at any time $t > 0$ is then incorporated into the previous knowledge (the set of instances available at time $t - 1$). For example, in $t = 1$ the current knowledge refers to that acquired in $t = 0$, and it is now employed to classify the first set of unlabelled patterns. After classifying, we edit the new labelled instances in order to discard possible misclassifications. Then the current knowledge is updated by including the instances that have not been eliminated in editing. The result will be the input set in $t = 2$.

The meaning of Alg1, Alg2, Alg3, and Alg4 in Tables 2, 3, 4 and 5 is as follows. In

the case of Alg1, we have employed the k-NCN algorithm for editing and the k-NN rule for classification. Alg2 uses the k-NCN algorithm both for editing and for classifying new patterns. Alg3 applies Wilson-Prob for editing and the k-Prob decision rule for classification. Finally, Alg4 is equal to Alg3, but using the nearest centroid neighborhood instead of the classical nearest neighborhood. Values in bold type indicate the first occurrence of the highest accuracy for each algorithm and each database.

t	Alg1	Alg2	Alg3	Alg4
0	66.67	66.67	**70.24**	68.45
1	66.07	66.07	70.24	69.05
2	66.07	68.45	69.64	69.64
3	66.07	69.64	67.86	69.64
4	65.48	69.05	67.26	69.64
5	65.48	69.05	67.86	69.64
6	66.07	69.64	67.86	69.64
7	66.67	70.24	67.86	70.24
8	**67.26**	**70.83**	67.86	**70.83**
9	67.26	68.45	66.67	70.24
1-NN	66.32			

Table 3: Classification accuracies for Diabetes database (1-NN indicates the classification accuracy when using the original TS without any editing).

From the results reported in Tables 2 and 3, some conclusions can be drawn. First, it has to be noted that all implementations outperform the 1-NN classification accuracy reported as a baseline. On the other hand, except Alg3 when applied over Diabetes and German databases, all the other cases show a certain improvement in performance with respect to the original edited TS ($t = 0$). Nonetheless, in terms of accuracy, it seems difficult to decide which learning algorithm is the best. In practice, any of those three algorithms (Alg1, Alg2, and Alg4) could constitute a good solution for increasing the knowledge in a partially supervised environment.

It is worth pointing out the fact that in general, the system obtains a maximum value in performance after processing a relatively small number of partitions. This is important because it can mean that after a number of iterations, the inclusion of more instances does not provide more information to the system. In this situation, the system increases the size of the TS, but without increasing its knowledge. This is a crucial issue that will be investigated in further extensions to this work.

t	Alg1	Alg2	Alg3	Alg4
0	67.61	67.61	**71.83**	69.01
1	69.01	69.01	71.83	69.01
2	**70.42**	**70.42**	71.83	69.01
3	70.42	70.42	71.83	69.01
4	70.42	70.42	70.42	69.01
5	70.42	70.42	67.61	**70.42**
6	67.61	70.42	67.61	70.42
7	67.61	70.42	69.01	70.42
8	67.61	70.42	69.01	70.42
9	67.61	70.42	69.01	70.42
10	67.61	70.42	70.42	70.42
11	67.61	70.42	70.42	70.42
12	67.61	70.42	70.42	70.42
1-NN	65.81			

Table 4: Classification accuracies for German database (1-NN refers to the classification accuracy when using the original TS without any editing).

t	Alg1	Alg2	Alg3	Alg4
0	51.61	51.61	54.84	54.84
1	61.29	58.06	58.06	61.29
2	61.29	**64.52**	**64.52**	61.29
3	64.52	64.52	64.52	64.52
4	**67.74**	64.52	64.52	64.52
5	67.74	64.52	64.52	64.52
6	67.74	64.52	64.52	64.52
7	67.74	64.52	64.52	**67.74**
1-NN	53.33			

Table 5: Classification accuracies for Heart database (1-NN refers to the classification accuracy when using the original TS without any editing).

5 Conclusions and further extensions

In this paper, a learning algorithm to increase the knowledge in partially supervised environments has been introduced. It makes use of a reduced number of labelled instances and a (possibly) large amount of unlabelled patterns. The system includes a set of tools allowing

to filter the new knowledge acquired during operation. Thus we pursue to avoid the risk of incorporating several mislabelled patterns into the TS and consequently, to degrade the overall system performance.

In the empirical evaluation of the learning system, we have used different classification rules and several editing algorithms. Except in the case of employing a scheme based on class conditional probabilities for both classification and editing (Alg3), all the other alternatives have been proven to perform well enough for increasing the knowledge.

An important issue related to the performance of a system with the capability of increasing its knowledge refers to the possibility for the TS size to grow too much and consequently, some problems related to storage space and classification time can make such a system useless. Although editing has the property, as a by-product, of reducing the TS size, this is not achieved in a considerable amount. Accordingly, possible extensions to this work are in the direction of including some techniques to intelligently reduce the TS size. To this end, both adaptive and selective condensing algorithms [12] can be of interest to control the TS size.

Also, the possibility of discovering new classes not present in the original TS can result important for this kind of learning systems in partially supervised domains. Therefore, future research includes the study of unsupervised techniques in order to incorporate this additional capability into our learning system.

Acknowledgments

This work has been partially supported by grant TIC2003-08496 from the Spanish CICYT (Ministerio de Ciencia y Tecnología).

References

[1] Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds, Machine Learning **56** (2004) 209–239.

[2] Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P.: Partially supervised clustering for image segmentation, Pattern Recognition **29** (1996) 859–871.

[3] Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts, In: Proc. 18th. Intl. Conf. on Machine Learning (2001) 19–26.

[4] Castelli, V., Cover, T.M.: On the exponential value of labeled samples, Pattern Recognition Letters **16** (1995) 105–111.

[5] Cover, T.M.: Estimation by the nearest neighbor rule, IEEE Trans. on Information Theory **14** (1968) 50–55.

[6] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification, IEEE Trans. on Information Theory **13** (1967) 21–27.

[7] Dasarathy, B.V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA (1991).

[8] Dasarathy, B.V.: Adaptive decision systems with extended learning for deployment in partially exposed environments, Optical Engineering **34** (1995) 1269–1280.

[9] Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs, NJ (1982).

[10] Ferri, F.J., Albert, J.V., Vidal, E.: Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules, IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics **29** (1999) 667–672.

[11] Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study, Intelligent Data Analysis **6** (2002) 429–449.

[12] Kim, S.-W., Oommen, B.J.: A brief taxonomy and ranking of creative prototype reduction schemes, Pattern Analysis & Applications **6** (2003) 232–244.

[13] Krogel, M.A., Scheffer, T.: Multirelational learning, text mining, and semi-supervised learning for functional genomics. Machine Learning **57** (2004) 61–81.

[14] Mantero, P., Moser, G., Serpico, S.B.: Partially supervised classification of remote sensing images through SVM-based probability density estimation, IEEE Trans. on Geoscience and Remote Sensing **43** (2005) 559–570.

[15] Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM, Machine Learning **39** (2000) 103–134.

[16] Okamoto, S., Yugami, N.: Effects of domain characteristics on instance-based learning algorithms, Theoretical Computer Science **298** (2003) 207–233.

[17] Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets, Pattern Recognition Letters **24** (2003) 1015–1022.

[18] Szummer, M.O.: Learning from Partially Labeled Data, PhD thesis, Massachusetts Inst. of Technology (2002).

[19] Tomek, I.: An experiment with the edited nearest neighbor rule. IEEE Trans. on Systems, Man and Cybernetics **6** (1976) 448–452.

[20] Vazquez, F., Sánchez J.S., Pla, F.: A stochastic approach to Wilson's editing algorithm, In: Pattern Recognition and Image Analysis, Lecture Notes in Computer Science **3523** (2005) 35–42.

[21] Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets, IEEE Trans. on Systems, Man and Cybernetics **2** (1972) 408–421.

Author Index