Universitat Autònoma de Barcelona

Coding and Decoding Design of ECOCs for Multi-class Pattern and Object Recognition

A dissertation submitted by **Sergio Escalera Guerrero** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, July 2008

Director: **Dr. Petia Radeva and Dr. Oriol Pujol**
Universitat de Barcelona
Dep. Matemàtica Aplicada i Anàlisi & Computer Vision Center

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 84-935251-8-7

Printed by Ediciones Gráficas Rey, S.L.

A mi abuela.

Acknowledgment

First of all, I thank Petia Radeva for the research education given during the years I have been doing my PhD. She is a research referent for me.

I also want to thank Oriol Pujol for his supervision and brainstorming capability that have done our work to be improved continuously.

My short stay in Delft was a very rich research period for me. I am very grateful to Robert P. W. Duin and David Tax for their supervision during those months. I learnt a lot from them. I am also grateful to the colleagues I met in Deft: Mauricio, Barat, Pavel, and Carmen.

I am also very grateful to the rest of my researching and teaching colleagues from the Computer Vision Center at the Universitat Autònoma de Barcelona. They made me feel as in a family during these years. I am not going to write the complete list of colleagues, but I am specially grateful to Àgata, Xevi, Poal, and Alicia.

Thanks also to my colleagues from the Universitat de Barcelona. I met them a few time ago but we are a nice group: Jordi Vitrià, Maria, Eloi, Jordi, Carles, Àngela, Inma, Jesús, and Maite.

I want to thank all my life friends and colleagues from university to have been part of my life: Dunia, Nuria, Eli, Merche, Garru, Igual, Mario, Jorge, Gomez, Max, Dreu, Ppl, Tania, Edgar, Alfred, Arantxa, Santi, Gallis, Trapis, etc.

Gracias mamá y papá por vuestro afecto y la educación que me habéis dado. Gracias hermano por estar ahí. Gracias a mis mujercitas Ana y Paula. Gracias abuela, siempre estás en mi corazón.

The last but not the least important acknowledgement goes to to the person who more suffered my good and bad moments. The person that gave me the necessary support to let me feel that all is possible, Mireia, my little heart.

Abstract

Many real problems require multi-class decisions. In the Pattern Recognition field, many techniques have been proposed to deal with the binary problem. However, the extension of many 2-class classifiers to the multi-class case is a hard task. In this sense, Error-Correcting Output Codes (ECOC) demonstrated to be a powerful tool to combine any number of binary classifiers to model multi-class problems. But there are still many open issues about the capabilities of the ECOC framework. In this thesis, the two main stages of an ECOC design are analyzed: the coding and the decoding steps. We present different problem-dependent designs. These designs take advantage of the knowledge of the problem domain to minimize the number of classifiers, obtaining a high classification performance. On the other hand, we analyze the ECOC codification in order to define new decoding rules that take full benefit from the information provided at the coding step. Moreover, as a successful classification requires a rich feature set, new feature detection/extraction techniques are presented and evaluated on the new ECOC designs. The evaluation of the new methodology is performed on different real and synthetic data sets: UCI Machine Learning Repository, handwriting symbols, traffic signs from a Mobile Mapping System, Intravascular Ultrasound images, Caltech Repository data set or Chaga's disease data set. The results of this thesis show that significant performance improvements are obtained on both traditional coding and decoding ECOC designs when the new coding and decoding rules are taken into account.

Resum

Molts problemes de la vida quotidiana estan plens de decisions multi-classe. En l'àmbit del Reconeixement de Patrons, s'han proposat moltes tècniques d'aprenentatge que treballen sobre problemes de dos classes. No obstant, la extensió de classificadors binaris al cas multi-classe és una tasca complexa. En aquest sentit, Error-Correcting Output Codes (ECOC) han demostrat ser una eina potent per combinar qualsevol nombre de classificadors binaris i així modelar problemes multi-classe. No obstant, encara hi ha molts punts oberts sobre les capacitats del framework d'ECOC. En aquesta tesis, els dos estats principals d'un disseny ECOC són analitzats: la codificació i la decodificació. Es presenten diferents alternatives de dissenys dependents del domini del problema. Aquests dissenys fan ús del coneixement del domini del problema per minimitzar el nombre de classificadors que permeten obtenir un alt rendiment de classificació. Per altra banda, l'anàlisi de la codificació de dissenys d'ECOC es emprada per definir noves regles de decodificació que prenen total avantatja de la informació provinent del pas de la codificació. A més a més, com que classificacions exitoses requereixen rics conjunts de característiques, noves tècniques de detecció/extracció de característiques es presenten i s'avaluen en els nous dissenys d'ECOC. L'avaluació de la nova metodologia es fa sobre diferents bases de dades reals i sintètiques: UCI Machine Learning Repositori, símbols manuscrits, senyals de trànsit provinents de sistemes Mobile Mapping, imatges coronàries d'ultrasò, imatges de la Caltech Repositori i bases de dades de malats de Chagas. Els resultats que es mostren en aquesta tesis mostren que s'obtenen millores de rendiment rellevants tant a la codificació com a la decodificació dels dissenys d'ECOC quan les noves regles són aplicades.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In real world problems, humans continuously base our behavior on making predictions based on previous knowledge. Obviously, not all of our predictions obtain the desired results, but our source of information, obtained from real life situations, makes us to make decisions with high **confidence** in many situations. Moreover, the more information available, the more confidence is achieved in our predictions. This process of recompiling information is clearly observable on babies. They are continuously looking, listening, touching their environment in order to make it familiar (fig. 1.1). In fact, what they are doing is to increase their source of information to be able to make good decisions in future situations. These decisions can be seen as a categorization among different **hypotheses** based on previous knowledge. This thesis deals with the problem of modelling multiple hypothesis to solve multi-class pattern and object categorization problems.

1.1 Short motivation of the thesis

Humans spend most of the time of our life learning from our environment, and reinforcing our knowledge in dreams. In this sense, learning is directly related with the human behavior. Nowadays, learning is used to help robotics to interact with their environment and to automatically solve problems without the need of human supervision. In this thesis, we focus on three research lines:

a) Multi-class categorization: Since the initials of Artificial Intelligence about 50 years ago, many learning techniques have been proposed to deal with many artificial systems. The initial learning designs were proposed to deal with just two classes. Which option is the best one given two previous possibilities? To solve this problem using several examples from two hypotheses, many learning techniques have been developed with successful results. However, in many real problems, it is common to face with problems where N possible solutions (where $N > 3$) exist.

Still, thought several powerful binary classifiers exist, each one of them focuses on different rules to model some types of problems, and one can not guarantee the use of only one of them to solve any type of classification tasks. In this sense, the study of new learning techniques is still an open issue. Moreover, though some state-of-

Figure 1.1: Visual perception.

the-art binary classifiers, such as Support Vector Machines or Adaboost, have been extended to deal with multi-class problems, the results in the multi-class case were so pessimistic. Because of this reason, it is common to conceive the classifiers to distinguish between just two classes and to combine them to solve multi-class tasks. Few combining strategies for binary classifiers have been proposed in the literature. In this sense, we claim to study robust multi-class classifiers in order to address the multi-class categorization task.

b) Object recognition: The object recognition problem in real images is a difficult task because of the high variability in appearance that objects suffer, such as illumination changes, partial oclusions, elastic and rigid deformations, etc. By this reason, we propose alternatives to the feature extraction and object modelling steps of the visual pattern recognition system, where the new multi-class categorization strategies could also be applied.

c) Real applications: Finally, an important issue of the thesis is to show the applicability and usefulness of the previous methods to solve real world multi-class object and pattern recognition problems. We proposed us to consider data from very different fields and prove that the developed methods and algorithms compete with the state-of-the-art methodologies for multi-class pattern and object recognition.

1.2 Statistical Pattern Recognition

Classification is the term in which some decision is made on the basis of a certain available information [27]. Thus, a classification procedure looks for a method that makes such judgements in new situations. In the Pattern Recognition community, the **binary classification** refers to make a decision for a new **object** ρ (**data sample** or **pattern**), classifying ρ by one of just two previous categories (or **classes**) c_1 and c_2. In the case of the binary classification, we have two alternatives to classify ρ, and thus, our decision is c_1 or c_2. More formally, given a **training data** $\{(\rho_1, \ell_1), .., (\rho_m, \ell_m)\}$,

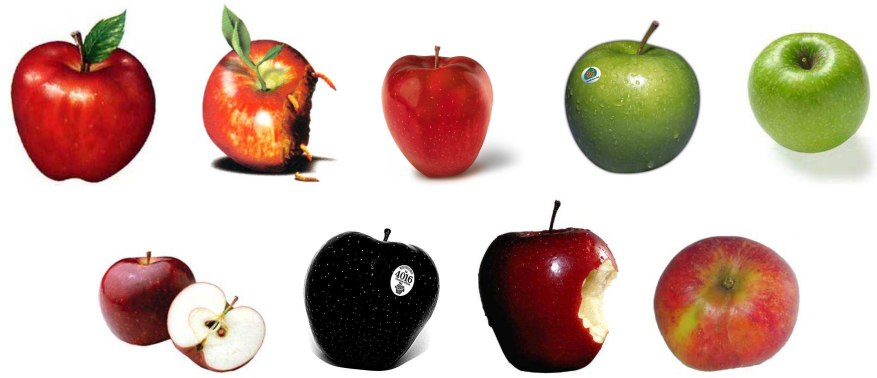


Figure 1.2: Apple samples.

where $\rho_i \in X$ is an object of **label** ℓ_i, where $\ell_i \in \{c_1, c_2\}$, a **classifier** h is **trained** to distinguish the objects of the set c_1 from the objects of the set c_2. The prediction is obtained in the form $h : X \rightarrow \ell$, where $\ell = \{\ell_1, \ell_2\}$. To **learn** the classifier h, each element ρ should be described with a set of characteristics (or **features**) inherent to the object. For example, we could describe a woman based on her age, height, weight, eyes color, etc. Then, the **learning** process uses the set of objects features from the two different classes to train the classifier h. An open question is which type of features should be used to describe a particular type of data.

There are many techniques that treats to deal with the problem of object **description** [49][10]. Representative features depend on the object and the problem one wants to solve. Moreover, some features can change their appearance when we observe the same object under different points of view, illumination changes, oclusions, rigid or elastic deformations, etc. The problem of object description is still a difficult task. Observe the objects of fig. 1.2. Which are the representative features to describe an apple? shape? color? Obviously, it depends on the categorization problem we consider. A binary classification not only consists on distinguishing apples from oranges. In the apple class, we can also apply categorization. Which apples have been bited? Which apples are red or green? These questions correspond to binary problems. In the first case, the variations of the shape of the object can be useful, but not the color. In the second case, the color has an outstanding decision, while the shape is not a relevant feature. Now, look at all apple samples. There is something wrong? An apple is dark! It is surprising for us. The shape corresponds to an apple, but we do not have previous knowledge about dark apples. Now, we are including information about black apples into our source of knowledge [75].

Given the features or measurements obtained for each pattern, in the statistical approach, each of these patterns is viewed as a point in a high-dimensional space. Then, the goal is to choose those features that allow pattern vectors belonging to different categories to occupy **compact** and **disjoint** regions in that **feature space** [38]. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated. Given a set of training patterns from each class, the objective is to establish **decision boundaries** in the feature

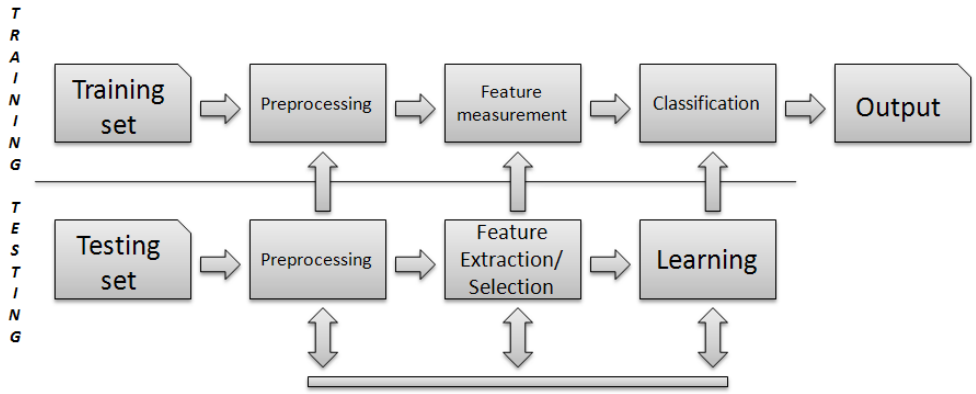


Figure 1.3: Statistical Pattern Recognition System.

space which separate patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which must either be specified or learnt [23][27].

The recognition system is operated in two modes: training (learning) and classification (testing) (see Fig. 1.3). The role of the preprocessing module is to segment the pattern of interest from the background, remove noise, normalize the pattern, and any other operation which will contribute in defining a compact representation of the pattern. In the training mode, the feature extraction/selection module finds the appropriate features for representing the input patterns and the classifier is trained to partition the feature space. The feedback path allows a designer to optimize the preprocessing and feature extraction/selection strategies. In the classification mode, the trained classifier assigns the input pattern to one of the pattern classes under consideration based on the measured features.

Based on the previous scheme, some points to design a robust Statistical Pattern Recognition model should be considered [28]:

How the data set should be designed?: This set has to be chosen such that it is representative for the set of objects to be recognized by the trained system.

How objects should be represented?: Real world objects have to be represented in a formal way in order to be analyzed and compared by mechanical means such as a computer. Moreover, the observations derived from the sensors or other formal representations have to be integrated with the existing, explicitly formulated knowledge either on the objects themselves or on the class they may belong to. The issue of representation is an essential aspect of pattern recognition and is different from classification. It largely influences the success of the final classification.

How the representation should be adapted to be learnt?: It is an intermediate stage between preprocessing and learning, in which representations, learning methodology or problem statement are adapted or extended in order to enhance the final recognition. This step may be neglected as being transparent, but its role is essential. It may reduce or simplify the representation, or it may enrich it by empha-

sizing particular aspects, e.g. by a nonlinear transformation of features that simplifies the next stage. Background knowledge may appropriately be (re)formulated and incorporated into a representation. If needed, additional representations may be considered to reflect other aspects of the problem. Exploratory data analysis (unsupervised learning) may be used to guide the choice of suitable learning strategies.

How can we generalize or infer?: At the learning stage, we learn a concept from a training set, the set of known and appropriately represented examples, in such a way that predictions can be made on some unknown properties of new examples. We either generalize towards a concept or **infer** a set of general rules that describe the qualities of the training data. The most common property is the class or pattern it belongs to, which corresponds to the classification task.

How the evaluation should be performed?: In this stage, we estimate how our system performs on known **training and validation data** while training the entire system. If the results are unsatisfactory, then the previous steps have to be reconsidered using the feedback module of fig. 1.3.

This thesis claims to focus on the previous questions to present powerful multi-class pattern and object recognition systems. We take into account the influence of the different types of feature sets that should be used to represent the objects from different types of problems. The embedding multi-class strategies that we present adapt the previous representation of the data, in a problem-dependent way, so that the learning process obtains high generalization performances.

1.3 Visual Pattern Recognition

One of the most challenging applications of statistical pattern recognition theory is the field of object recognition in images. Many of the visual Pattern Recognition techniques presented in the literature are biological inspired [79]. The idea in selective attention is that not all parts of an image give us information and analyzing only the relevant parts of the image in detail is sufficient for recognition and classification. The biological structure of the eye is such that a high resolution fovea and its low-resolution periphery provide data for recognition purposes. The fovea is not static, but is moved around the visual field in sacades. These sharp, directed movements of the fovea are not random. The periphery provides low-resolution information, which is processed to reveal **salient points** as targets for the fovea, and those are inspected with the fovea. The eye movements are a part of overt attention, as opposed to covert attention which is the process of moving an attentional 'spotlight' around the perceived image without moving the eye. In the case of Neural Networks, the objective is to simulate the behavior of some neuronal circuits of our brain.

To model a Visual Pattern Recognition problem, a common approach consists of detecting the objects in an image, and then, classifying them to their respective category. Many recognition systems also treat the problem of object detection as a binary classification problem, where the information of each part of the image is classified as object or background. Look to the situation presented in fig. 1.4. The robot Aibo of Sony captures images from a scene, discarding background regions. At the first step, Aibo treats to find the regions of the image that contain a fruit.

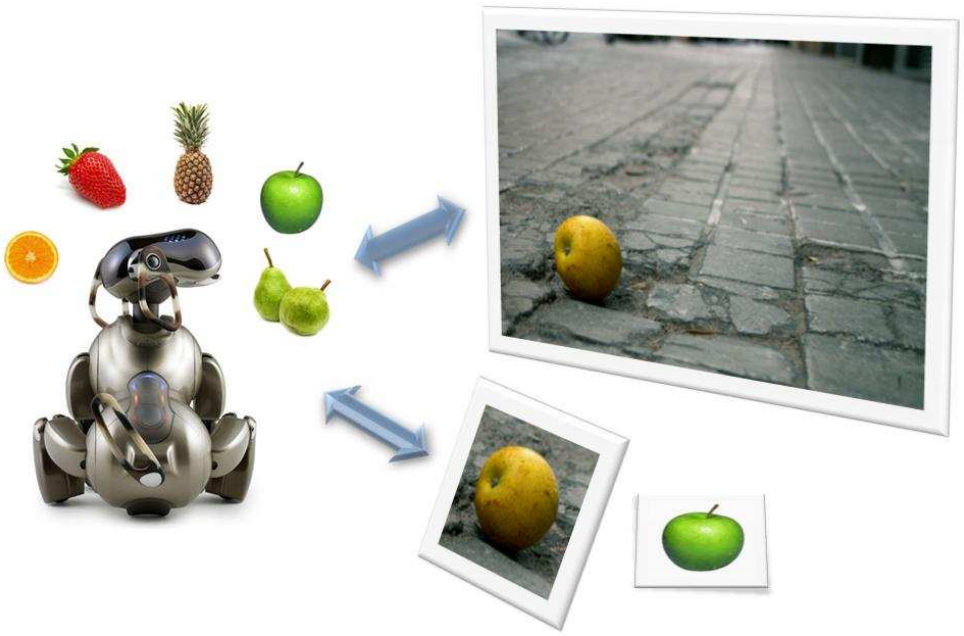


Figure 1.4: Aibo detects and classifies a fruit in an scene.

Once the region containing a fruit is found, given five previous fruit categories, Aibo classifies the inner object as an apple. Hence, the problem of **object recognition** can be seen as an **object detection** problem followed by a **classification procedure**.

The previous steps of Visual Pattern Recognition are shown in the scheme of fig. 1.5. An example of each step for a face detection application is shown. First, at the *keypoint detection* step some regions of the images that could belong to face features are detected. Then, all these regions are processed at the *description* module in order to analyze their content. Based on the interpretation of the features of each previous detected keypoint or **landmark**, a selection or **clustering** is applied in order to discard **false positive** regions that do not belong to face features. Finally, the selected and described regions are included in a model, such a correlogram in the example, in order to learn the parts and structure of the face instance. Note that these modules correspond to the preprocessing and feature extraction/selection stages of the Statistical Pattern Recognition module of fig. 1.3.

1.4 The Multi-class Categorization Problem

When we talk about binary classification, the labels from classes c_1 and c_2 use to take the values +1 or -1, respectively. At the learning process explained above, the labels for the training objects are known. This is called **Supervised Learning** [27]. There are situations where we have a set of observations and our aim is to establish the existence of classes or **clusters** in the data. It is called **Unsupervised Learning** or

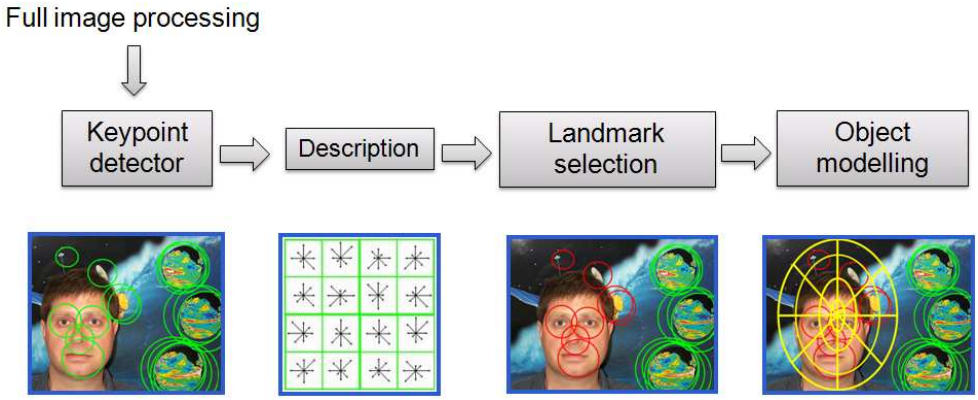


Figure 1.5: Pre-processing biological-inspired techniques for Visual Pattern Recognition.

clustering [27].

At the previous examples, the problem corresponds to Supervised Learning for Binary Classification. But in real-world problems, we do not only have binary classification. We can classify between red apples and green apples or we can also distinguish among apples, oranges, bananas, pears, etc. **Multi-class classification** is the term applied to those machine learning problems that require assigning labels to instances where the labels are drawn from a set of at least three classes. Real-world situations are full of multi-class classification problems, where we want to distinguish among N possible classes or hypotheses (obviously, the number of objects that we have learnt during our life tends to be uncountable). If we can design a multi-classifier h, then the prediction can be understood as in the binary classification problem, being $h : X \rightarrow \ell$, where now $\ell = \{\ell_1, .., \ell_N\}$, for a N-class problem. Several multi-class classification strategies have been proposed in the literature. However, though there are very powerful binary classifiers, many strategies fail to manage multi-class information. As we show later, a possible multi-class solution consists of designing and combining of a set of binary classification problems.

1.5 Classifiers

In the Statistical Pattern Recognition field, classifiers are frequently grouped into those based on similarities, probabilities, or geometric information about class distribution [38].

a) Similarity Maximization Methods: The Similarity Maximization Methods use the similarity between patterns to decide a classification. The main issue in this type of classifiers is the definition of the similarity measure.

b) Probabilistic Methods: The most well known probabilistic methods make use of Bayesian Decision Theory. The decision rule assigns class labels to that having the **maximum posterior probability**. The posterior can be calculated by the

well-known Bayes rule:

$$posterior = \frac{likelihood \times prior}{evidence} \tag{1.1}$$

If $P(c_i)$ is the **prior probability** that a given instance ρ belongs to class c_i, $p(\rho|c_i)$ is the **class-conditional probability density function**: the density for ρ given that the instance is of class c_i, and $p(\rho)$ is defined as $\sum p(\rho|c_j) \times P(c_j)$ over all classes. Then, eq. 1.1 is equivalent to:

$$P(c_j|\rho) = \frac{p(\rho|c_j) \times P(c_j)}{p(\rho)} \tag{1.2}$$

The classification is done in favor of the j^{th} class is $P(c_j|\rho) > P(c_i|\rho)$, $\forall c_i \in C$ and $c_i \neq c_j$, where C is the set of classes ($c_j \in C$).

c) Geometric Classifiers: Geometric classifiers build decision boundaries by directly minimizing the error criterion.

Table 1.1 summarizes the main classification strategies studied in literature. For each strategy, we show its properties, comments, and type based on the previous grouping.

Table 1.1: Classification methods.

Method	Property	Comments	Type
Template matching	Assigns patterns to the most similar template	The templates and the metric have to be supplied by the user; the procedure may include nonlinear normalizations; scale (metric) dependent	Similarity Maximization
Nearest Mean Classifier	Assigns patterns to the nearest class mean	No training needed; fast testing; scale (metric) dependent	Similarity Maximization
Subspace Method	Assigns patterns to the nearest class subspace	Instead of normalizing on invariants, the subspace of the invariant is used; scale (metric) dependent	Similarity Maximization
1-Nearest Neighbor Rule	Assigns patterns to the class of the nearest training pattern	No training needed; robust performance; slow testing; scale (metric) dependent	Similarity Maximization
k-Nearest Neighbor Rule	Assigns Patterns to the majority class among k nearest neighbor using a performance optimized value for k	Asymptotically optimal; scale (metric) dependent, slow testing	Similarity Maximization
Bayes plug-in	Assigns pattern to the class which has the maximum estimated posterior probability	Yields simple classifiers (linear or quadratic) for Gaussian distributions; sensitive to density estimation errors	Probabilistic
Logistic Classifier	Maximum likelihood rule for logistic (sigmoidal) posterior probabilities	Linear classifier; iterative procedure; optimal for a family of different distributions (Gaussian); suitable for mixed data types	Probabilistic
Parzen Classifier	Bayes plug-in rule for Parzen density estimates with performance optimized kernel	Asymptotically optimal; scale (metric) dependent; slow testing	Probabilistic
Fisher Linear Discriminant	Linear classifier using MSE optimization	Simple and fast; similar to Bayes plug-in for Gaussian distributions with identical covariance matrices	Geometric
Binary Decision Tree	Finds a set of thresholds for a pattern-dependent sequence of features	Iterative training procedure; overtraining sensitive; needs pruning; fast testing	Geometric
Adaboost	Logistic regression for a combination of weak classifiers	Iterative training procedure; overtraining sensitive; fast training; good generalization performance	Geometric
Perceptron	Iterative optimization of a linear classifier	Sensitive to training parameters; may produce confidence values	Geometric
Multi-layer Perceptron (Feed-Forward Neural Network)	Iterative MSE optimization of two or more layers of perceptrons (neurons) using sigmoid transfer functions	Sensitive to training parameters; slow training; nonlinear classification function; may produce confidence values; overtraining sensitive; needs regularization	Geometric
Radial Basis Network	Iterative MSE optimization of a feed-forward neural network with at least one layer of neurons using Gaussian-like transfer functions	Sensitive to training parameters; nonlinear classification function; may produce confidence values; overtraining sensitive; needs regularization; may be robust to outliers	Geometric
Support Vector Classifier	Maximizes the margin between the classes by selecting a minimum number of support vectors	Scale (metric) dependent; iterative; slow training; nonlinear; overtraining insensitive; good generalization performance	Geometric

1.5.1 Multi-class classifiers

There are plenty of classification techniques reported in literature for the multi-class problem: Support Vector Machines, decision trees, nearest neighbors rules, etc. Most of the state-of-the-art classification strategies (see table 1.1) are defined to deal with 2-class problems. Strategies that obtain good generalization performance in the 2-class case, such as Adaboost or Support Vector Machines, have been extended to the multi-class case, but this extension is not always trivial. In such cases, the usual way to proceed is to reduce the complexity of the problem into a set of simpler binary classifiers and combine them. An usual way to combine these simple classifiers is the **voting scheme**.

Voting (or **averaging**) is a technique that, instead of using the best hypothesis learnt so far, uses a weighted average of all hypotheses learnt during a training procedure. The averaging procedure is expected to produce more stable models, which leads to less **overfitting**. Some multi-class combining techniques use different classification strategies to split sub-sets of classes and model the classification problem as a combination of different types of decision boundaries in a voting scheme. In this thesis, we focus on the combination of classifiers where the **base classifier** for each individual classification problem of the ensemble is based on the same type of decision boundary. Next, we briefly review the standard voting schemes used in the literature.

One Versus the Rest

To get a N-class classifier, it is common to construct a set of binary classifiers $\{h_1, .., h_N\}$, each one trained to split one class from the rest of classes, and use the outputs of each binary classifier to predict one of the N classes [68].

Pairwise Classification or One Versus One

In pairwise classification, we train a classifier for each possible pair of classes [88]. For N classes, this results in $N(N-1)/2$ binary classifiers. This number is usually larger than the number of one-versus-the-rest classifiers; for instance, if $N = 10$, one needs to train 45 binary classifiers rather than 10 as in the one-versus-the-rest strategy. Although this suggests larger training times, the individual problems that we need to train on are significantly smaller, and if the training algorithm scales superlinearly with the training set size, it is actually possible to save time.

Similar considerations apply to the runtime execution speed. When one try to classify a test pattern, we evaluate all 45 binary classifiers, and classify according to which of the classes gets the highest number of votes. A vote for a given class is defined as the classifier putting the pattern into that class. The individual classifiers, however, are usually smaller in size than they would be in the one-versus-the-rest approach. This is for two reasons: First, the training sets are smaller, and second, the problems to be learnt are usually easier, since the classes have less overlap.

Error-Correcting Output Codes

It is known that for some classification problems, the lowest error rate is not always reliably achieved by trying to design a single classifier. An alternative approach is to use a set of relatively simple sub-optimal classifiers and to determine a combination strategy that pools together the results. Different types of systems of multiple classifiers have been proposed in the literature, most of them use similar constituent classifiers, which are often called base classifiers.

Error-Correcting Output Codes (ECOC) were born as a general framework to combine binary problems to address the multi-class problem. The strategy was introduced by Dietterich and Bakiri [24] in 1995. Based on the error correcting principles [24], ECOC has been successfully applied to a wide range of applications, such as face recognition [94], face verification [44], text recognition [33] or manuscript digit classification [100].

The ECOC technique can be broken down into two distinct stages: encoding and decoding. Given a set of classes, the coding stage designs a codeword[1] for each class based on different binary problems. The decoding stage makes a classification decision for a given test sample based on the value of the output code.

At the coding step, given a set of N classes to be learnt, n different bi-partitions (groups of classes) are formed, and n binary problems (**dichotomizers**) are trained. As a result, a codeword of length n is obtained for each class, where each bit of the code corresponds to the response of a given dichotomizer (coded by +1, -1, according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1, 1\}^{N \times n}$ in the binary ECOC framework. This binary strategy has been extended in several research directions showing promising results. Allwein et al. [5] improved the representability of the ECOC technique by adding a third symbol to the coding matrix. With this new symbol each element of the coding matrix is chosen from $\{-1, 0, +1\}$, where classes with zero value are not considered for that particular dichotomizer. Figure 1.6 shows an example of a ternary ECOC configuration. The white regions correspond to +1, the black regions to -1, and the grey regions to the zero symbol. Four classes are codified in this example, obtaining a codeword for each class (rows of the coding matrix). Then, each of the columns corresponds to a binary problem, where the +1 positions are the classes for the first group of a classifier, and the -1 positions of the column correspond to the classes of the second group of a classifier, with a total of six binary problems in this example.

The ternary ECOC allows to express the classic pairwise and one-versus-all schemes in a common framework as well as to define new coding strategies such as random dense or random sparse output codes. Most of these coding ECOC strategies are defined independently of the problem domain or the classification performance. The first approach to ECOC coding design was proposed by Utschick et al. [91]. In their work, they optimize a maximum-likelihood objective function by means of the expectation maximization algorithm in order to improve the process of binary coding. As mentioned by the authors "the results of some experiments make us believe that for

[1]The codeword is a sequence of bits of a code representing each class, where each bit identifies the membership of the class for a given binary classifier.

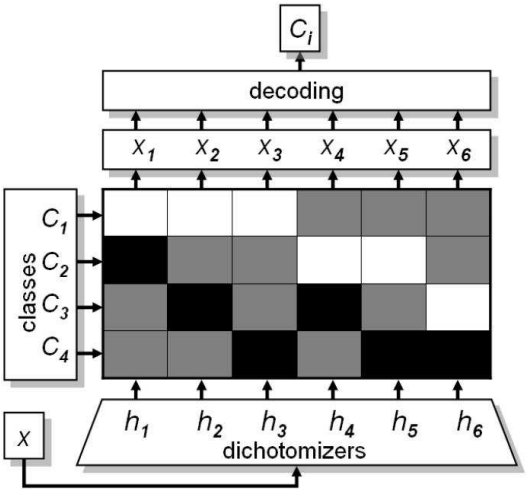


Figure 1.6: Example of an ECOC configuration.

many polychotomous classification problems, the OPC [one-versus-all] method is still the optimal choice for the output coding. Crammer et al. [17] also reported improvement in the design of the ECOC codes. However, their results were rather pessimistic since they proved that the problem of finding the optimal discrete codes is computationally unfeasible. As an alternative, they proposed a method for heuristically search of the optimal coding matrix by relaxing the representation of the code matrix from discrete to continuous values. Recently, new improvements in the problem dependent coding techniques have been presented by Pujol et al. [73]. In their work, the authors proposed the embedding of discriminant tree structures derived from the problem domain in the ECOC framework. As a result, they obtained a compact discrete coding matrix with a small number of dichotomizers and very high accuracy. In this point, an open question is how to design a problem-dependent ECOC matrix so that it simultaneously minimizes the code length meanwhile maximizing the generalization capability of the ensemble.

The decoding step was originally based on error-correcting principles under the assumption that the learning task can be modelled as a communication problem, in which class information is transmitted over a channel [24]. During the decoding process, applying the n binary classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class defined in the matrix M, and the data point is assigned to the class with the *closest* codeword. Concerning the decoding strategies, two of the most standard techniques are the Euclidean distance and the Hamming decoding distance. If the minimum Hamming distance between any pair of class codewords is d, then any $[(d-1)/2]$ errors in the individual dichotomies result can be corrected in the binary ECOC framework, since the nearest codeword will be the correct one. In the previous example of fig. 1.6, the six binary problems $\{h_1, .., h_6\}$ test a new data sample. Each classifier introduce a +1 or -1 value at each corresponding codeword position, and a decoding strategy is

applied. Finally, the input test sample is classified by the class c_i with the closest decoding value. In the case that we decode a ternary symbol-based coding matrix, three possible symbols are considered, and an open question is if the traditional 2-symbol-based decoding strategies can still be used in the ternary case.

Multi-class built-in classifiers

Two of the most frequently multi-class classifiers used in the literature are extensions of the binary Adaboost and SVM. The multi-class variant of Adaboost that has demonstrated to be dominant to the other proposals in empirical studies is the Adaboost.MH [82]. The Adaboost.MH algorithm converts the N-class problem into that of estimating a two-class classifier on a training set N times as large, with an additional feature defined by the set of class labels [101]. It can be seen as an one-versus-all scheme, representing a "Multi-label Hamming" to measure the classification prediction, being essentially the one-versus-all ECOC design with Hamming decoding. Moreover, in [42], the authors showed that the behavior of the multi-class support vector machines and fuzzy support vector machines also tend to the performance of the one-versus-all ECOC strategy. Recently, Torralba et.al. [90] proposed a novel multiclass approach where instead of training independent classifiers for each object class, they are jointly trained leading to a more robust feature extraction and better recognition generalization. As we will show, the previous schemes correspond to the simple ECOC designs. Thus, a more exhaustive analysis of the ECOC capabilities can be very suitable to deal with multi-class categorization problems, such as visual pattern recognition tasks.

1.6 State-of-the-art on Visual Pattern Recognition

Concerning to the model of fig. 1.5 for visual pattern recognition, several approaches have been proposed in the literature. We showed that this scheme is an instance of the statistical pattern recognition system shown in fig. 1.3. There are many examples of visual pattern recognition applications that can be found in real life: in the case of optical character recognition, the goal is to find the digit value or the character letter. In object recognition, a new instance is categorized according to the pool of trained objects (cars, motorbikes, horses, flowers, etc.). In medical imaging, for instance, a potential application would be the automatic classification of different kind of plaque tissues (lipidic, fibrous, calcified, necrotic, etc.).

Usually, the problem of object recognition of visual pattern recognition systems (e.g. person identification) needs a previous detection of the object class category (e.g. face location). Object detection is concerned with the reliable and accurate location of target objects in an image. In general, according to the way objects are described, three main families of approaches can be considered [59]: Part-based, Patch-based, and Region-based methods.

a) Part-based approaches consider that an object is defined as a specific spatial arrangement of its parts fragments. Following this idea, an efficient Bayessian network for learning the spatial arrangement of parts is proposed in [84]. An unsupervised statistical learning of constellation of parts and spatial relations is used

in [30]. Other authors [36] propose to use Attribute Relational Graphs for describing spatial relations. In [6] a representation integrating Boosting with constellations of contextual descriptors is defined. In this work, the feature vector includes the bins that correspond to the different positions of the correlograms determining the object properties.

b) Another family of recognition techniques is the **Patch-based** methods, which classify each rectangular image region of a fixed aspect ratio (shape) at multiple sizes, as object (or parts of the target object) or background. In this topic, the authors of [4] use a dictionary of parts and a window algorithm for learning active features of the object are proposed. A similar technique is found in [90], where objects are described by the best features obtained using masks and normalized cross-correlation.

c) Finally, **Region-based** algorithms segment regions of the image from the background and describe them by a set of features that provide texture and shape information.

The selection of feature points can be based on image contour points [6] or other image features, as for example provided by interest point detectors. The point detectors have been used in multiple applications: matching for stereo pairs [61], image retrieval from large databases [83], object retrieval in video [87], shot location [81], and object categorization [29], to mention just a few. One of the most well-known keypoint detector is the Harris detector [54]. The method is based on searching for edges that are maintained at different scales to detect interest image points. Several variants and applications based on the Harris point detector have been used in the literature, such as Harris-Laplacian [35], Affine variants [54], DoG [49], etc. Finally, the models obtained by the visual pattern recognition system should be recognized using some kind of classification technique, as the ones mentioned before.

1.7 Contribution

Error-Correcting Output Codes were proposed to deal with multi-class problems by embedding several binary problems in a coding matrix. This approach showed to be very robust applied to many real-world problems. However, several aspects of this framework that can help us to improve the classification performance have not been previously analyzed. In this thesis, we theoretically and empirically analyze either the binary as the ternary ECOC frameworks.

1) ECOC Coding: Concerning the coding step of the ECOC framework, we propose different alternatives to deal with problem-dependent coding designs of Error-Correcting Output Codes. As we show, problem-dependent designs are capable to model difficult problems with a reduced number of classifiers using the knowledge of the problem domain:

1.1) Forest-ECOC: One important point of the ECOC technique is that the information provided by the classifiers of the ensemble is combined to obtain a classification prediction. We take advantage of this property to propose the embedding of multiple tree structures in an ECOC matrix. In this way, the internal nodes of the trees share their information to robustly classify new data samples.

1.2) ECOC-ONE: The optimizing node embedding ECOC technique is proposed

to extend any initial coding matrix. The training data is split into training and validation sub-sets in order to search the optimal bi-partitions of classes which trained classifier minimizes the training error meanwhile avoids overtraining. A greedy search and a modified sequential forward floating search are proposed to look for the best dichotomizers at each step of the procedure.

1.3) Sub-class ECOC: Moreover, we present a coding strategy to split an original N-class problem into a N'-class problem (so that $N' > N$) in order to define multi-class data easier to be learnt, avoiding overtraining and being able to model overlapping data. Classifiers that can not model multi-class problems based on the initial distributions of the data are able to fit decision boundaries over the new split partitions of classes.

2) ECOC Decoding: Concerning the decoding step of the ECOC framework, we present a new taxonomy common to all decoding strategies. The state-of-the-art decoding strategies are evaluated on the new representation, and different alternatives to decode are proposed to deal with a successful classification either in the binary or ternary ECOC frameworks.

2.1) Attenuated Euclidean distance: This technique is proposed to avoid the influence of the ECOC coding matrix positions that do not provide relevant information of the data.

2.2) Laplacian decoding: The Laplacian decoding introduces a measure that counts the number of coincidences between the input codeword and the class codeword, normalizing by the total number of codeword positions. The procedure introduces a previous bias to make the technique robust in cases of having a small number of coded positions in one word.

2.3) Pessimistic β-Density Distribution decoding: This technique estimates the probability density functions between two codewords. The main goal of this strategy is to model at the same time the accuracy and uncertainty based on a pessimistic score on the continuous binomial distribution in order to obtain more reliable predictions.

2.4) Loss-weighted decoding: The Loss-Weighted decoding strategy codifies a matrix of weights that ponders the decoding process. This matrix avoids the influence of the positions that do not provided information at the coding step. At same time, the technique makes the decoding measures between codewords comparable either in the binary as in the ternary ECOC framework.

3) Sparse ECOC designs: The influence of the decoding analysis presented in the thesis also suggests the re-definition of some coding strategies. In particular, we show that the definition of the state-of-the-art Sparse random ECOC matrix is inconsistent, and we propose a new measure of ternary error-correction capability and ternary codeword separability.

4) Object detection and description: Concerning the visual pattern recognition system, we propose new techniques for the preprocessing and feature extraction/selection modules.

4.1) Boosted Landmarks of contextual descriptors: We propose a technique for generic object detection problems. Objects are described by constellations of features, where Adaboost learns at the same time the relevant object features and their spatial arrangement.

4.2) Blurred Shape Models: We propose a technique for object description. The

technique focuses on relevant object shape points to codify its spatial arrangement.

These object detection and description techniques are also applied with the previous multi-class ECOC classification techniques to solve real pattern recognition problems.

5) Real applications: All the previous methods are evaluated on several real applications with real and synthetic data, such as UCI Machine Learning Repository data sets, real traffic signs, intravascular tissue images, handwriting data sets, Caltech repository, and Chaga's disease data set.

The experimental results of this thesis show that the presented strategies outperform the results of the state-of-the-art ECOC coding and decoding designs as well as the state-of-the-art multi-classifiers, being specially suitable to model several real multi-class categorization problems.

1.8 Thesis Outline

The thesis is organized as follows: Section 2 gives the fundamentals of the coding and decoding steps of the ECOC framework either in the binary and the ternary framework. Section 3 presents the new designs for problem-dependent ECOC coding. Three problem-dependent designs are presented: The Forest-ECOC, the ECOC Optimum Node Embedding, and the Sub-class ECOC approach. Section 4 presents a new taxonomy of decoding strategies and four alternatives to decode are proposed. Section 5 presents a new analysis for the separability of ternary codes for sparse designs. Section 6 describe the novel techniques for object detection and description. A list of applications based on the new methodology are presented in section 7. Finally, section 8 concludes the thesis.

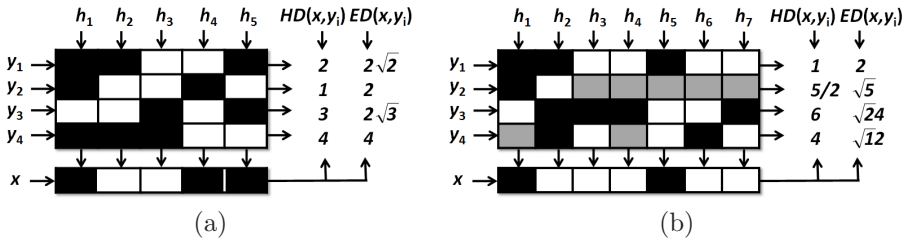
In the appendices, we summarize the notation used for the ECOC analysis and describe some techniques used by the present methodology. Last appendix contains the publications regarding the content of the thesis.

Chapter 2

Error-Correcting Output Codes

Given a set of N classes to be learnt in an ECOC framework, n different bi-partitions (groups of classes) are formed, and n binary problems (dichotomizers) over the partitions are trained. As a result, a codeword of length n is obtained for each class, where each position (bit) of the code corresponds to a response of a given dichotomizer (coded by +1 or -1 according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1, +1\}^{N \times n}$ in the binary case. In fig. 2.1(a) we show an example of a binary coding matrix M. The matrix is coded using 5 dichotomizers $\{h_1, ..., h_5\}$ for a 4-class problem $\{c_1, ..., c_4\}$ of respective codewords $\{y_1, ..., y_4\}$. The hypotheses are trained by considering the labeled training data samples $\{(\rho_1, l(\rho_1)), ..., (\rho_m, l(\rho_m))\}$ for a set of m data samples. The white regions of the coding matrix M are coded by +1 (considered as one class for its respective dichotomizer h_j), and the dark regions are coded by -1 (considered as the other one). For example, the first classifier is trained to discriminate c_3 against c_1, c_2, and c_4; the second one classifies c_2 and c_3 against c_1 and c_4, etc., as follows:

$$h_1(x) = \begin{cases} 1 & \text{if } x \in \{c_3\} \\ -1 & \text{if } x \in \{c_1, c_2, c_4\} \end{cases}, \ldots, \quad h_5(x) = \begin{cases} 1 & \text{if } x \in \{c_2, c_4\} \\ -1 & \text{if } x \in \{c_1, c_3\} \end{cases} \tag{2.1}$$



(a) (b)

Figure 2.1: (a) Binary ECOC design for a 4-class problem. An input test codeword x is classified by class c_2 using the Hamming or the Euclidean Decoding. (b) Example of a ternary matrix M for a 4-class problem. A new test codeword x is classified by class c_1 using the Hamming and the Euclidean Decoding.

During the decoding process, applying the n binary classifiers, a code x is obtained for each data sample ρ in the test set. This code is compared to the base codewords $(y_i, i \in [1, .., N])$ of each class defined in the matrix M. And the data sample is assigned to the class with the *closest* codeword. In fig. 2.1(a), the new code x is compared to the class codewords $\{y_1, ..., y_4\}$ using the Hamming [68] and the Euclidean Decoding [5]. The test sample is classified by class c_2 in both cases, correcting one bit error.

In the ternary symbol-based ECOC, the coding matrix becomes $M \in \{-1, 0, +1\}^{N \times n}$. In this case, the symbol zero means that a particular class is not considered for a given classifier. A ternary coding design is shown in fig. 2.1(b). The matrix is coded using 7 dichotomizers $\{h_1, ..., h_7\}$ for a 4-class problem $\{c_1, ..., c_4\}$ of respective codewords $\{y_1, ..., y_4\}$. The white regions are coded by 1 (considered as one class by the respective dichotomizer h_j), the dark regions by -1 (considered as the other class), and the grey regions correspond to the zero symbol (classes that are not considered by the respective dichotomizer h_j). For example, the first classifier is trained to discriminate c_3 against c_1 and c_2 without taking into account class c_4, the second one classifies c_2 against c_1, c_3, and c_4, etc. In this case, the Hamming and Euclidean Decoding classify the test data sample by class c_1. Note that a test codeword can not contain the zero value since the output of each dichotomizer is $h_j \in \{-1, +1\}$.

The analysis of the ECOC error evolution has demonstrated that ECOC corrects errors caused by the bias and the variance of the learning algorithm [25][1]. The variance reduction is to be expected, since ensemble techniques address this problem successfully and ECOC is a form of voting procedure. On the other hand, the bias reduction must be interpreted as a property of the decoding step. It follows that if a point ρ is misclassified by some of the learnt dichotomies, it can still be classified correctly after being decoded due to the correction ability of the ECOC algorithm. Non-local interaction between training examples leads to different bias errors. Initially, the experiments in [25] show the bias and variance error reduction for algorithms with *global* behavior (when the errors made at the output bits are not correlated). After that, new analysis also shows that ECOC can improve performance of *local* classifiers (e.g., the k-nearest neighbor, which yields correlated predictions across the output bits) by extending the original algorithm or selecting different features for each bit [78].

2.1 Coding designs

In this section, we review the state-of-the-art on coding designs. We divide the designs based on their membership to the binary or the ternary ECOC frameworks.

2.1.1 Binary coding

The standard binary coding designs are the one-versus-all [68] strategy and the dense random strategy [5]. In one-versus-all, each dichotomizer is trained to distinguish one

[1]The bias term describes the component of the error that results from systematic errors of the learning algorithm. The variance term describes the component of the error that results from random variation and noise in the training samples and random behavior of the learning algorithm. For more details, see [25].

class from the rest of classes. Given N classes, this technique has a codeword length of N bits. An example of an one-versus-all ECOC design for a 4-class problem is shown in fig. 2.2(a). The dense random strategy generates a high number of random coding matrices M of length n, where the values $\{+1, -1\}$ have a certain probability to appear (usually $P(1) = P(-1) = 0.5$). Studies on the performance of the dense random strategy suggested a length of $n = 10 \log N$ [5]. For the set of generated dense random matrices, the optimal one should maximize the Hamming Decoding measure between rows and columns (also considering the opposites), taking into account that each column of the matrix M must contain the two different symbols $\{-1, +1\}$. An example of a dense random ECOC design for a 4-class problem and five dichotomizers is shown in fig. 2.2(b). The complete coding approach was also proposed in [5]. Nevertheless, it requires the complete set of classifiers to be measured $(2^{N-1} - 1)$, which usually is computationally unfeasible in practice.

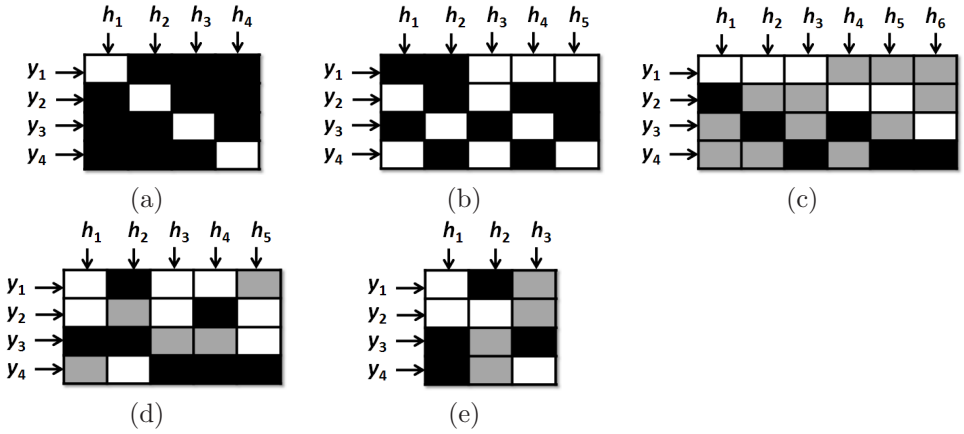


Figure 2.2: Coding designs for a 4-class problem: (a) one-versus-all, (b) dense random, (c) one-versus-one, (d) sparse random, and (e) DECOC.

2.1.2 Ternary Coding

The standard ternary coding designs are the one-versus-one strategy [88] and the sparse random strategy [5]. The one-versus-one strategy considers all possible pairs of classes, thus, its codeword length is of $\frac{N(N-1)}{2}$. An example of an one-versus-one ECOC design for a 4-class problem is shown in fig. 2.2(c). The sparse random strategy is similar to the dense random design, but it includes the third symbol zero with another probability to appear, given by $P(0) = 1 - P(-1) - P(1)$. Studies suggested a sparse code length of $15 \log N$ [5]. An example of a sparse ECOC design for a 4-class problem and five dichotomizers is shown in fig. 2.2(d). In the ternary case, the complete coding approach can also be defined.

Due to the huge number of bits involved in the traditional coding strategies, new problem-dependent designs have been proposed [91][17][73]. The new techniques are based on exploiting the problem domain by selecting the representative binary

problems that increase the generalization performance while keeping the code length small. The Discriminant ECOC (DECOC) of [73] is based on the embedding of discriminant tree structures derived from the problem domain. The binary trees are built by looking for the sub-sets of classes that maximizes the mutual information between the data and their respective class labels. As a result, the length of the codeword is only $(n-1)$. The algorithm is summarized in table 2.1. In fig. 2.3, a binary tree structure for an 8-class problem is shown. Each node of the tree splits a sub-set of classes. Each internal node is embedded in the ECOC matrix as a column, where the white regions correspond to the classes on the left sub-sets of the tree, the black regions to the classes on the right sub-sets of the tree, and the grey regions correspond to the non-considered classes (set to zero). Another example of a DECOC design for a 4-class problem obtained by embedding a balanced tree is shown in fig. 2.2(e).

Table 2.1: DECOC algorithm.

DECOC: Create the Column Code Binary Tree as follows:

Initialize L to $L_0 = \{\{c_1, .., c_N\}\}$

while $|L_k| > 0$

1) Get $S_k : S_k \in L_k, k \in [0, N-2]$

2) Find the optimal binary partition $BP(S_k)$ that maximizes the fast quadratic mutual information [73].

3) Assign to the column t of matrix M the code obtained by the new partition $BP(S_k) = \{C_1, C_2\}$.

4) Update the sub-sets of classes L_k to be trained as follows:

$L_k^{'} = L_k \backslash S_k$
$L_{k+1} = L_k^{'} \cup C_i$ iff $|C_i| > 1, i \in [1, 2]$

It can be seen that increasing the number of classes leads to increasing the number of classifiers. Table 2.2 and fig. 2.4 summarize the cost in terms of the number of binary classifiers required for the binary and ternary coding strategies.

2.2 Decoding designs

In this section, we review the state-of-the-art on decoding designs. The decoding strategies (independently of the rules they are based on) are divided depending if they were designed to deal with the binary or the ternary ECOC frameworks.

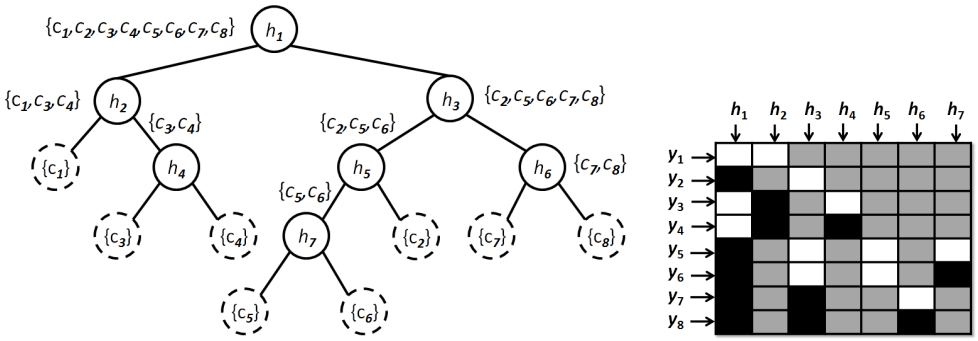


Figure 2.3: Example of a binary tree structure and its DECOC codification.

Table 2.2: Number of dichotomizers required for each coding design.

Coding design	Number of classifiers
DECOC	$N-1$
one-versus-all	N
dense random	$10 \log N$
sparse random	$15 \log N$
one-versus-one	$\frac{N(N-1)}{2}$
complete binary	$2^{N-1}-1$
complete ternary	$\sum_{i=2}^{N} \binom{N}{i} (2^{i-1}-1)$

2.2.1 Binary decoding

The binary decoding designs most frequently applied are: Hamming Decoding [68], Inverse Hamming Decoding [95], and Euclidean Decoding [5].

• *Hamming Decoding*

The initial proposal to decode is the Hamming Decoding measure. This measure is defined as follows:

$$HD(x, y_i) = \sum_{j=1}^{n} (1 - sign(x^j y_i^j))/2 \tag{2.2}$$

This decoding strategy is based on the error correcting principles under the assumption that the learning task can be modeled as a communication problem, in which class information is transmitted over a channel, and two possible symbols can be found at each position of the sequence [24].

• *Inverse Hamming Decoding*

The Inverse Hamming Decoding [95] is defined as follows: let Δ be the matrix composed by the Hamming Decoding measures between the codewords of M. Each

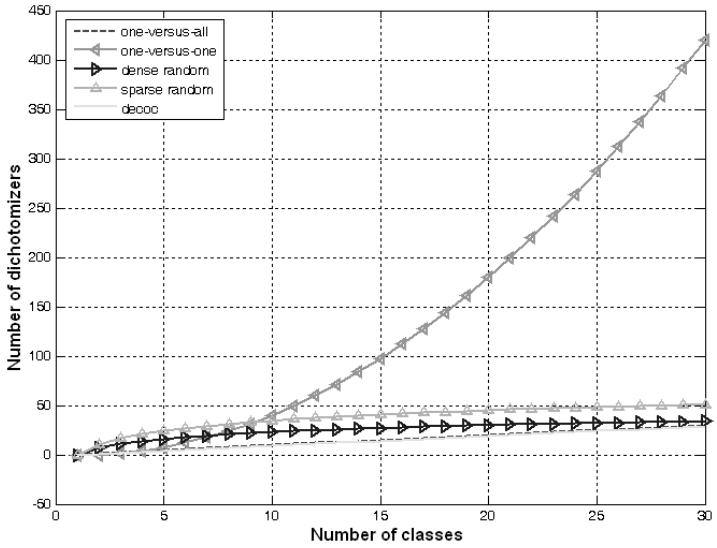


Figure 2.4: Number of classifiers required for the coding strategies when the number of classes increases.

position of Δ is defined by $\Delta(i_1, i_2) = HD(y_{i_1}, y_{i_2})$. Δ can be inverted to find the vector containing the N individual class likelihood functions by means of:

$$IHD(x, y_i) = max(\Delta^{-1} D^T) \tag{2.3}$$

where the values of $\Delta^{-1} D^T$ can be seen as the proportionality of each class codeword in the test codeword, and D is the vector of Hamming Decoding values of the test codeword x for each of the base codewords y_i. The practical behavior of the IHD showed to be very close to the behavior of the HD strategy [68].

• *Euclidean Decoding*

Another well-known decoding strategy is the Euclidean Decoding. This measure is defined as follows:

$$ED(x, y_i) = \sqrt{\sum_{j=1}^{n} (x^j - y_i^j)^2} \tag{2.4}$$

2.2.2 Ternary decoding

Concerning the ternary decoding, the state-of-the-art strategies are: Loss-based Decoding [5], and the Probabilistic Decoding [67].

• *Loss-based Decoding*

The Loss-based Decoding strategy [5] chooses the label ℓ_i that is most consistent with the predictions f (where f is a real-valued function $f : \rho \rightarrow R$), in the sense

that, if the data sample ρ was labeled ℓ_i, the total loss on example (ρ, ℓ_i) would be minimized over choices of $\ell_i \in \ell$, where ℓ is the complete set of labels. Formally, given a Loss-function model, the decoding measure is the total loss on a proposed data sample (ρ, ℓ_i):

$$LB(\rho, y_i) = \sum_{j=1}^{n} L(y_i^j f^j(\rho)) \tag{2.5}$$

where $y_i^j f^j(\rho)$ corresponds to the *margin* and L is a Loss-function that depends on the nature of the binary classifier. The two most common Loss-functions are $L(\theta) = -\theta$ (Linear Loss-based Decoding (LLB)) and $L(\theta) = e^{-\theta}$ (Exponential Loss-based Decoding (ELB)). The final decision is achieved by assigning a label to example ρ according to the class c_i which obtains the minimum score.

• *Probabilistic Decoding*

Recently, the authors of [67] proposed a probabilistic decoding strategy based on the continuous output of the classifier to deal with the ternary decoding. The decoding measure is given by:

$$PD(y_i, F) = -log \left(\prod_{j \in [1,\ldots,n]: M(i,j) \neq 0} P(x^j = M(i,j)|f^j) + K \right) \tag{2.6}$$

where K is a constant factor that collects the probability mass dispersed on the invalid codes, and the probability $P(x^j = M(i,j)|f^j)$ is estimated by means of:

$$P(x^j = y_i^j|f^j) = \frac{1}{1 + e^{y_i^j(v^j f^j + \omega^j)}} \tag{2.7}$$

where vectors v and ω are obtained by solving an optimization problem [67].

2.3 ECOC discussion

In this chapter, we reviewed the state-of-the-art on coding and decoding Error-Correcting Output Codes strategies. Most of the coding strategies described above are problem-independent, either in the binary as well as in the ternary ECOC framework, and can be used over any kind of multi-class problem without redefine the coding matrix. In those cases, to take full benefit from the ECOC capabilities, we should generate large codewords, such as in the case of the random designs, so that we can fit a proper decision boundaries for a multi-class problem even in case of not using previous information of the problem-domain.

We also reviewed recent approaches that model an ECOC matrix based on the information of each particular problem-domain. In this case, the training step spends more time, but the codeword length is smaller than previous approaches meanwhile the generalization capability of the system tends to be increased.

On the other hand, we reviewed the state-of-the-art ECOC decoding strategies proposed in the literature to deal either with the binary as well as the ternary ECOC decoding. We showed that, though there exist several binary decoding strategies, a widely used approach to decode binary coding matrices is the Hamming decoding. We showed that the use of a third symbol in the ECOC matrix made to define new decoding strategies able to work with the information provided by the new symbol.

However, as we show on the next chapters, the ECOC coding step can still benefit from the use of the knowledge of the problem-domain. Moreover, the extension from binary to ternary decoding is not a trivial problem, and a deep analysis is provided to deal with a successful decoding.

Chapter 3

ECOC Coding: Problem-Dependent ECOC designs

Most of the ECOC coding strategies presented in the literature are defined independently of the problem domain or the classification performance. In fact, very little attention has been paid in literatures to the coding process of the ECOC matrix.

In this chapter, we present three problem-dependent coding designs. First, we take advantage of the Discriminant ECOC approach [73] to design a Forest-ECOC, where a set of tree structures defined in a problem-dependent way are embed in an ECOC design. Second, we present a guided procedure of ECOC where the binary problems are selected so that the learning error decreases in a training and validation sets. Finally, we propose a sub-class approach of ECOC. The Sub-class design splits the original data into sub-classes, being able to model difficult problems that can not be solved for a given base classifier using the original set of classes.

3.1 Forest-ECOC

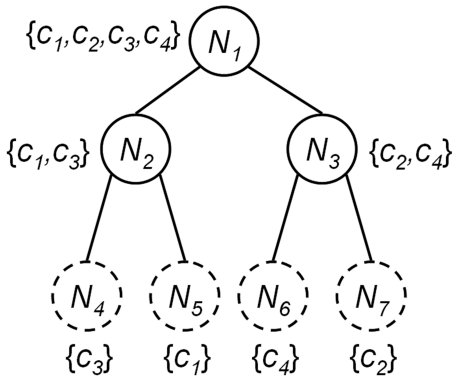
In the work of Pujol et al. [73], a method for embedding tree structures in the ECOC framework is proposed. The mutual information is used to estimate the sub-partitions of classes of each node of the tree. Beginning on the root containing all classes, the nodes associated to the best partition in terms of the mutual information are found, and the process is repeated until the sets with a single class are obtained. The main advantage of this method is that it achieves a successful classification performance at the same time that it maintains the code length small.

Taking the previous work as a baseline, we propose to use multiple trees embedding, forming a Forest-ECOC. We build an optimal tree - the one with the highest classification score at each node - and several suboptimal trees - the ones closer to the optimal one under certain conditions. Let us keep at each iteration the best k partitions of the set of classes. If the best partition is used to construct the current ECOC tree, the rest of partitions form the roots of $k-1$ trees. We repeat iteratively this process until all nodes from the trees are decomposed into one class. Given a base classifier, the sub-optimal tree candidates are designed to have the maximum classification score at each node without repeating previous sub-partitions of classes. In the case of generating T first optimal trees, we can create an ensemble of trees by embedding them in the ECOC matrix, as shown in Algorithm 3.1.

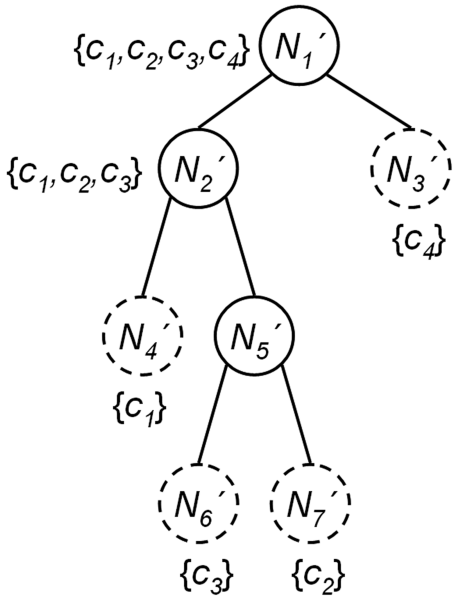
The proposed technique provides a sub-optimal solution because of the combination of robust classifiers obtained from a greedy search using the classification score. One of the main advantages of the proposed technique is that the trees share their information among classes in the ECOC matrix M. It is done at the decoding step by considering all the coded positions of a class jointly instead of separately. It is easy to see that each tree structure of N classes introduces $N-1$ classifiers, that is far from the $\frac{N(N-1)}{2}$ dichotomizers required for the one-versus-one coding strategy. An example of two optimal-trees and the Forest-ECOC matrix for a toy problem is shown in Figure 3.1. The Figure 3.1(a) and (b) show two examples of optimal trees. The second optimal tree is constructed based on the following optimal sub-partitions of classes. In this way, for the first initial set of classes $\{c_1, c_2, c_3, c_4\}$, the two optimal trees include the best sub-partitions of classes in terms of the classification score, that in the example corresponds to c_1, c_3 vs c_2, c_4 for the first tree, and c_1, c_2, c_3 vs c_4 for the second tree, respectively. Figure 3.1(c) shows the embedding of trees into the Forest-ECOC matrix M. Note that the column h_3 corresponds to the node N_3, and the following dichotomizers correspond to the nodes of the second tree. The classes that do not belong to the sub-partitions of classes are set to zero. On the other hand, the classes belonging to each partition are set to $+1$ and -1 values, defining the subset of classes involved on each classifier.

Given an input sample to test with the Forest-ECOC matrix, we obtain the Forest-ECOC vector where each vector component is the result of solving each binary classifier trained on each of the columns of the matrix. Note that this procedure can be cast in the multi-task framework since it combines knowledge from different binary problems and shares their knowledge among the tasks.

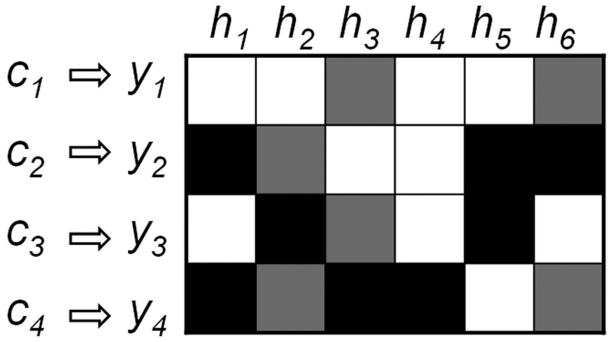
The second step of the ECOC process is the decoding. Here we can apply any



First optimal tree for a four-class problem



Second optimal tree for the same problem



Forest-ECOC matrix M for the problem, where h_1, h_2 and h_3 correspond to classifiers of N_1, N_2 and N_3 from the first tree, and h_4, h_5 and h_6 to N_1', N_2' and N_5' from the second tree

Figure 3.1: Four-class optimal trees and the Forest-ECOC matrix.

Table 3.1: Training algorithm for the Forest-ECOC.

Given N classes: $c_1, ..., c_N$ and T trees to be embedded

for $t = 1, .., T$ do

Initialize the tree root with the set $N_0 = \{c_1, ..., c_N\}$

$i \Leftarrow 0$

Generate the best tree at iteration t:

for each node N_i do

Train the best partition of its set of classes $\{C_1 C_2\}|N_i = C_1 \cup C_2$ using a classifier h_i (if the sub-partition has not been previously considered) so that the training error is minimal

end for

According to the partition obtained at each node, codify each column of the matrix M as:

$$M(r,i) = \begin{cases} 0 & \text{if } c_r \notin N_i \\ +1 & \text{if } c_r \in C_1 \\ -1 & \text{if } c_r \in C_2 \end{cases}$$

where r is the index of the corresponding class c_r

$i \Leftarrow i + 1$

end for

decoding strategy presented on next chapters to decode a Forest-ECOC design.

3.1.1 Forest-ECOC Evaluation

In order to validate the accuracy of the Forest-ECOC we tested it on the UCI Machine Learning repository repository [8].[1]

UCI Evaluation

The compared methods are: 40 runs of multiclass joint boosting with decision stumps [90][2], all pairs Fisher Linear Discriminant Analysis (FLDA) with a previous 99.9% of the Principal Components Analysis, and Dense Random ECOC. Our method and

[1] More experimental results and analysis of the Forest-ECOC methodology are shown in chapter 7.

[2] Multi-class joint boosting is a relatively new multi-class approach where instead of training independent classifiers for each object class, they are jointly trained. This training is performed by finding common features that can be shared across classes, leading to a robust feature extraction and a good generalization of the recognition problem.

Table 3.2: Classification results for UCI data sets.

UCI	JB	all pairs FLDA	Forest ECOC	Dense ramdom ECOC
Yeast	**56.54±1.42**	52.32±1.65	**53.85±1.64**	47.32±0.93
Dermathology	**96.14±0.92**	**96.40±1.33**	**95.32±1.31**	**96.57±0.74**
Ecoli	**85.50±1.06**	**84.62±1.92**	**83.98±1.13**	81.15±1.55
Segmentation	92.83±1.01	86.81±0.91	**94.98±0.66**	73.89±0.56
Satimage	**80.02±1.18**	**81.92±1.92**	73.91±1.11	72.85±0.83
Vowel	64.86±1.74	74.28±1.37	**77.67±1.81**	41.32±1.38
Pendigits	90.22±0.69	**93.94±2.35**	81.42±1.93	78.41±1.44
Rank	1.57	1.57	**1.42**	3.0

Dense Random ECOC use Gentle Adaboost with decision stumps as a classification technique to estimate the Forest-ECOC dichotomizers, with $T = 2$ to generate and embed multiple trees and the Attenuated Euclidean Decoding method of section to decode. The state-of-the-art in random strategies are the dense random and the sparse random coding techniques. As the dense random strategy tends to improve the classification rate of the sparse case for the same number of binary problems [73], we tested this strategy with the same number of dichotomizers as our Forest-ECOC approach. The probability of appearance of the $\{1, -1\}$ is 0.5 in both cases, so we tested 10000 matrices to obtain the one that maximizes the row and column Hamming distance [5].

Looking at the results in table 3.2 we can observe that our method is competitive with the three commented approaches, and it attains the first position in the classification ranking for 8 UCI data sets (the details of the UCI data sets can be found in chapter F). The table shows the mean accuracy using stratified ten-fold cross-validation, and the confidence interval at 95% using a two tailed t-test. The ranking has been obtained considering that all techniques with results overlapping with the confidence interval of the top performance technique are considered also as first choice. Observe that the Forest-ECOC compares favorably to the other approaches; in this sense, it turns out a promising technique for the purposes of multi-class recognition.

3.2 ECOC Optimum Node Embedding

The ECOC-ONE technique is motivated by the necessity of having fast algorithms with high discriminative power able to generate as much as necessary number of dichotomizers in order to obtain the desired performance. The work of [73] has motivated the look for techniques with small codeword length that provide high performance in general conditions. In this section, we propose a general procedure to increase the accuracy of any ECOC coding by adding very few optimal dichotomizers. In this sense, if the original coding has small length, the extension after the ECOC-ONE results in a still compact codewords but with increased performance. In particular, we apply this technique to optimize the initial embedded tree proposed in [73].

3.2.1 ECOC-ONE definition

ECOC-Optimal Node Embedding defines a general procedure capable of extending any coding matrix by adding dichotomizers based on a discriminability criterion. In the case of a multiclass recognition problem, our procedure starts with a given ECOC coding matrix. We increase this ECOC matrix in an iterative way, adding dichotomizers that correspond to different sub-partitions of classes. These partitions are found using greedy optimization based on the confusion matrices so that the ECOC accuracy improves on both training and validation sets. The training set guides the convergence process, and the validation set is used to avoid overfitting and to select a configuration of the learning procedure that maximizes the generalization performance. Since not all problems require the same dichotomizers structure -in form of sub-partitions-, our optimal node embedding approach generates an optimal ECOC-ONE matrix dependent on the hypothesis performance in a specific problem domain.

3.2.2 Optimizing node embedding

In order to explain our procedure, we divide the ECOC-ONE algorithm in 6 steps: optimal tree generation, weights estimation, accuracy estimate based on confusion matrix, defining the new optimal dichotomizer, and ECOC matrix M construction.

Let us define the notation used in the following paragraphs: given a data pair (ρ, l), where s is a multidimensional data point and l is the label associated to that sample, we define $S = \{(\rho, \mathbf{l})\} = \{(\rho_\mathbf{t}, \mathbf{l_t})\} \bigcup \{(\rho_\mathbf{v}, \mathbf{l_v})\}$, where $S_t = \{(\rho_\mathbf{t}, \mathbf{l_t})\}$ and $S_v = \{(\rho_\mathbf{v}, \mathbf{l_v})\}$ are the sets of data pairs associated to training and validation sets, respectively. In the same way, $e(h(\rho), \mathbf{l})$ represents the empirical error over the data set ρ given an hypothesis $h(\cdot)$.

a) Optimal tree generation

We propose the use of a binary tree structure using accuracy as a sub-partition splitting criterion. This proposal differs from the one in [73] that uses the mutual information to form the nodes, without taking into account the particularities of the current classification scheme. We initialize the root of the tree with the set containing all the classes. Afterwards, for the tree building, each node of the tree is generated by

an exhaustive search[3] of the sub-partition of classes associated to the parent node, so that the classifier using that sub-partition of classes attains maximal accuracy on the training and validation subsets. In fig. 3.2, the sub-partition of classes required at each node of the optimal tree is shown. For example, given the root node containing all classes, the optimal partition achieving the least error is given by $\{\{c_1 \cup c_3\}, \{c_2 \cup c_4\}\}$. Once we have generated the optimal tree, we embed each internal node of the tree into the coding matrix M in the following way: consider the partition of the set of classes associated to a node $C = \{C_1 \cup C_2 | C_1 \cap C_2 = \oslash\}$. The element (i, r) of the ECOC-ONE matrix corresponding to class i and dichotomizer r is given by:

$$M(i,r) = \begin{cases} 0 & \text{if } c_i \notin C \\ +1 & \text{if } c_i \in C_1 \\ -1 & \text{if } c_i \in C_2 \end{cases} \tag{3.1}$$

Although, this strategy is the one chosen in this article for our initial coding, note that any coding could be used instead[4].

b) Weights estimates

It is known that when a multiclass classification problem is decomposed into binary problems, not all of these base classifiers have the same importance. In this way, our approach introduces a weight to adjust the importance of each dichotomizer in the ensemble ECOC matrix. In particular, the weight associated to each column depends on the error when applying the ECOC to both training sets (training and validation) in the following way,

$$w_i = 0.5 \log\left(\frac{1 - e(h_i(\rho), l)}{e(h_i(\rho), l)}\right) \tag{3.2}$$

where w_i is the weight for the i^{th} dichotomizer, and $e(h_i(\rho), l)$ is the error produced by this dichotomizer at the affected classes on both sets of the partition. This equation is based on the weighted scheme of the additive logistic regression [32]. In the following section, we explain how we select the dichotomizers and how their weights affect the convergence of the algorithm.

c) Test accuracy of the training and validation sets

Once constructed the binary tree and its corresponding coding matrix, we look for additional dichotomizers in order to focus on the examples that are difficult to classify. To select the next optimal node, we test the current M accuracy on S_t and S_v resulting in a_t and a_v, respectively. We combine both accuracies in the following way[5]:

$$a_{total} = \frac{1}{2}(a_t + a_v)$$

[3] In the case that the number of classes makes the exhaustive computation unfeasible we can use SFFS as explained in [73].

[4] In the discussion section, the reader can find the results of the application of our extension technique using the one-versus-all strategy as initial coding.

[5] Other combinations are possible, but we consider that the importance of the validation set must be very significant when compared to the training accuracy. Otherwise, the total accuracy will have a major influence of the training set and the benefit from the validation set will be minimal. Moreover, we have experimentally observed that this combination leads in general to slightly better results than other split criteria.

In order to find each accuracy value, we obtain the resulting codeword $x \in \{-1, 1\}^n$ using the strong hypothesis $\mathcal{H} = \{h_1, ..., h_j\}$ for each sample of these sets, and we label it as follows:

$$\tilde{\mathrm{l}} = argmin_j \quad \big(d(x, y_j)\big) \tag{3.3}$$

where $d(\cdot)$ is a distance estimation between codeword x and the codeword y_j. $\mathcal{H}(M, h, \rho)$ is the strong hypothesis resulted from the application of the set of learning algorithms $h(\cdot)$ on the problems defined by each column of the ECOC matrix M on a data point ρ. The result of $\mathcal{H}(M, h, \rho)$ is an estimated codeword x. We propose to use the Attenuated Euclidean Decoding presented in section 4.3.

d) The training and validation confusion matrices

Once we test the accuracy of the strong hypothesis $\mathcal{H}$ on S_t and S_v, we estimate their respective confusion matrices $\nu_t(\mathbf{S_t})$ and $\nu_v(\mathbf{S_v})$. Both confusion matrices are of size $N \times N$, and have at position (i, j) the number of instances of class c_i classified as class c_j.

$$\nu_k(i, j) = \mid \{(\rho, l)_k : l = c_i, h(\rho) = c_j\} \mid, k = \{t, v\} \tag{3.4}$$

where l is the label estimation. Once the matrices have been obtained, we select the pair $\{c_i, c_j\}$ with maximal value according to the following expression:

$$\{c_i, c_j\} = argman_{c_i, c_j; i \neq j}\big(\nu_t(i, j) + \nu_t^T(i, j) + \nu_v(i, j) + \nu_v^T(i, j)\big) \tag{3.5}$$

$\forall (i, j) \in [1, ..., N]$, where ν^T is the transposed matrix of ν. The resulting pair is the set of classes that are most easily confounded, and therefore they have the maximum partial empirical error

e) Find the new dichotomizer

Once the set of classes $\{c_i, c_j\}$ with maximal error has been obtained, we create a new column of the ECOC matrix. Each candidate column considers a possible subpartition of classes $\wp = \{\{\{c_i\} \bigcup C_1\}, \{\{c_j\} \bigcup C_2\}\} \subseteq C$ so that $C_1 \cap C_2 \cap c_i \cap c_j = \oslash$ and $C_i \subseteq C$. In particular, we are looking for the subset division of classes $\wp$ so that the dichotomizer h_t associated to that division minimizes the empirical error defined by $e(\mathcal{H}(\rho), l)$.

$$\tilde{\wp} = argmin_{\wp} \quad \big(e(\mathcal{H}(\rho), l)\big) \tag{3.6}$$

Once defined the new sets of classes, the column components associated to the set $\{\{c_i\}, C_1\}$ are set to $+1$, the components of the set $\{\{c_j\}, C_2\}$ are set to -1 and the positions of the rest of classes are set to zero. In the case that multiple candidates obtain the same performance, the one involving more classes is preferred. Firstly, it reduces the number of uncertainty in the ECOC matrix by reducing the number of zeros in the dichotomizer. Secondly, one can see that when more classes are involved, the generalization achieved is greater. Each dichotomizer finds a more complex rule on a greater number of classes. This fact has also been observed in the work of Torralba et al. [90]. In their work, a multi-task scheme is presented that yields to a classifier with an improved generalization by aids of class grouping algorithms.

f) Update the matrix

The column m_i is added to the matrix M and its weight w_i is calculated using equation (3.2).

Table 3.3: ECOC-ONE general algorithm

Given N_c classes and a coding matrix M:

while $error > \varepsilon$ or $error_t < error_{t-1}$, $t \in [1, T]$:

Compute the optimal node t:

1) Test accuracy on the training and validation sets S_t and S_v.

2) Select the pair of classes $\{c_i, c_j\}$ with the highest error analyzing the confusion matrices from S_t and S_v.

3) Find the partition $\wp_t = \{C_1, C_2\}$ that minimizes the error rate in S_t and S_v.

4) Compute the weight for the dichotomizer of partition $\wp_i$ based on its classification score.

Update the matrix M.

Table 3.3 shows the summarized steps for the ECOC-ONE approach. Note that, the process described is iterated while the error on the training subsets is greater than ε or the number of iterations $i \leq T$[6].

3.2.3 Sub-optimal embedding

When the number of classes is high enough, exhaustive search optimization is computationally unfeasible. In this case, the problem should be addressed using a modification of the sequential forward floating search.

Pudil et al. in [70] introduced a family of suboptimal search algorithms called *floating search methods* effective in high dimensional problems involving non-monotonic search criteria. This method was proposed as a suboptimal search method for alleviating the prohibitive computation cost of exhaustive search strategies in feature selection. This family of methods is directly related to the *plus-l take away-r* algorithm. However, the first 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 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 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.

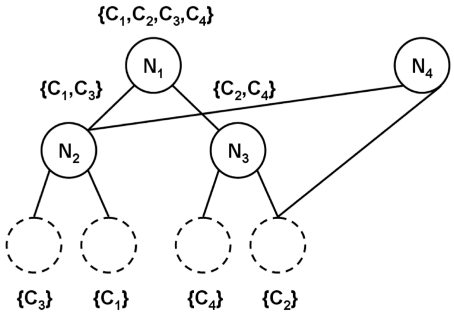
The algorithm used in this paper is a modified version of the top-down approach called Sequential Forward Floating Search (MSFFS, see table 3.4). The most notable difference from the SFFS is that we work with three sets of elements: a pool of elements Y and the two searched sets X^1, X^2. In this case, both sets start empty $X_0^1 = X_0^2 = \emptyset$ and they are filled from the pool set while the search criterion J applied to both sets increases. The most beneficial item from the pool of elements is added to the corresponding set at each inclusion step. In the conditional exclusion step, the worst item from both sets is removed if the criterion keeps increasing. In our approach, the criterion used for designing this partition is the empirical error. In the context of our ECOC problem, the two sets $\{X^1, X^2\}$ are the sub-partition sets of classes $\wp_t = \{C_1, C_2\}$.

3.2.4 ECOC-ONE example

An example of an ECOC-ONE strategy applied to a four-class classification example can be found in figure 3.2. The initial optimal tree corresponds to the dichotomizers

[6]The stopping criterion of our method involves two cases: Firstly, the case in which the combined error is reduced to zero. If both training and validation errors go to zero the method should stop because we can not obtain meaningful information from now on. Therefore, ϵ is usually set to zero unless some *a priori* knowledge about the acceptable error is considered. Second, since the sub-optimal node embedding tries to increase the accuracy of the ECOC coding increasing the number of bits per word, a certain number of bits should be decided to be the maximum allowable for our application. In our experiments, T is usually set to values in the range $[2 \dots N]$, where N is the number of classes. We selected this range of values in order to increase the global performance with very few additional dichotomizers.

of optimal sub-partition of the classes. This tree has been generated using accuracy as a sub-partition splitting criterion. After testing the performance of the ensemble tree (composed by the columns $\{h_1, h_2, h_3\}$ of the ECOC matrix M of fig. 3.2(b)), let assume that classes $\{c_2, c_3\}$ get maximal error in the confusion matrices ν_t and ν_v. We search for the sub-partition of classes using the training and validation subsets so that the error between $\{c_2, c_3\}$ and all previous misclassified samples is minimized. Suppose now that this sub-partition is $\{c_1, c_3\}$ versus $\{c_2\}$. As a result, a new node N_4 corresponding to dichotomizer h_4 is created. We can observe in fig. 3.2 that N_4 uses a class partition that is present in the tree. In this sense, this new node connects two different nodes of the tree. Note that using the previously included dichotomizers, the partition $\{c_1, c_3\}$ is solved by N_2. In this way, the Hamming distance between c_2 and c_3 is increased by adding the new dichotomizer to the whole structure. At the same time, the distance among the rest of the classes is usually maintained or slightly modified.



(a)

	h_1	h_2	h_3	h_4
C_1	1	-1	0	1
C_2	-1	0	-1	-1
C_3	1	1	0	1
C_4	-1	0	1	0

(b)

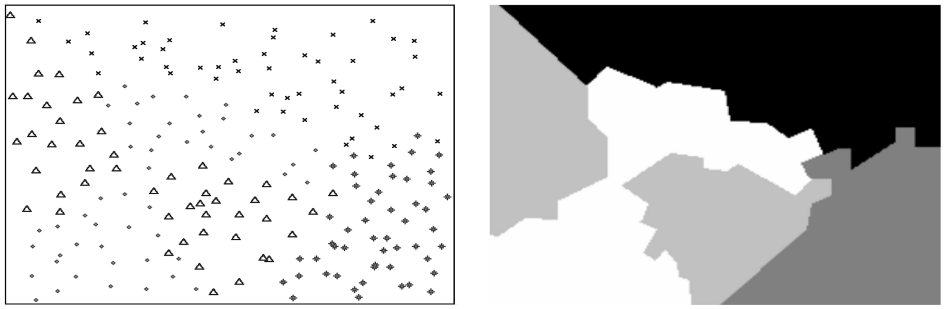
Figure 3.2: (a) Optimal tree and first optimal node embedded, (b) ECOC-ONE code matrix M for four dichotomizers.

As mentioned before, one of the desirable properties of the ECOC matrix is to have maximal distance between rows. Our procedure focuses on the relevant difficult partitions, increasing the distance between "close" classes. This fact improves the robustness of the method since difficult classes are likely to have a greater number of dichotomizers centered on them. In this sense, it creates different geometrical arrangements of decision boundaries, and leads the dichotomizers to make different bias errors.

3.2.5 ECOC-ONE in a 4-class toy problem

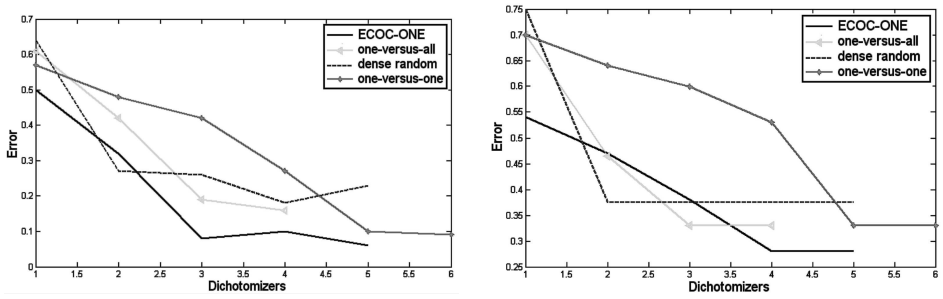
To analyze the properties of our proposed technique and compare it to the state-of-art approaches, we have designed the toy classification problem of fig. 3.3(a). This multiclass problem has 50 samples for each of the four classes. The ideal boundaries are shown in fig. 3.3(b). In this particular case, two of the classes are difficult to classify (triangles and dots). The number of dichotomizers used in this toy problem, for each ECOC technique, are: 6 for one-versus-one, 4 for one-versus-all, and 5 for Dense Random and ECOC-ONE. We select 5 dichotomizers for the ECOC-ONE and

the Dense-random technique because we want to show the performance when the number of hypothesis is smaller than the one-versus-one method. An illustration of the training evolution process for all the techniques is shown in fig. 3.4(a) where the error is given as a function of the number of dichotomizers. One can observe a greater error reduction for ECOC-ONE with few dichotomizers compared to the rest of methods. The test evolution for the same problem is shown in fig. 3.4(b), where the number of dichotomizers and the error rate are shown at x-axis and y-axis, respectively. Figure 5 displays two ECOC matrices used in this evaluation: ECOC-ONE (M_{one}) with column weights (W_{one}) and Dense Random (M_{dense}).



(a) (b)

Figure 3.3: (a) 4 classes for a toy problem, (b) classes boundaries for the toy problem



(a) (b)

Figure 3.4: (a) Train evolution for the toy problem. (b) Test evolution for the toy problem.

Table 3.5 shows the ten-fold cross-validation results for all the commented ECOC techniques. In this table, the accuracy, the confidence interval at 95%, and the number of dichotomizers used are displayed. The results on this toy classification problem show that our technique outperforms the others. An example of the trained boundaries for all the techniques at one iteration of cross-validation is shown in fig. 3.6. The dark lines correspond to the real boundaries and the grey regions to the learning errors. We can observe that the regions of ECOC-ONE (fig. 3.6(a)) are better defined. Note that two different Dense Random matrices with the same distance create

$$\mathbf{W_{one}} = \begin{pmatrix} 2 & 2 & 2 & 0.9229 & 1.0271 \end{pmatrix}$$

$$\mathbf{M_{one}} = \begin{pmatrix} -1 & -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \text{(a)}$$

$$\mathbf{M_{dense}} = \begin{pmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 \end{pmatrix} \quad \text{(b)}$$

Figure 3.5: ECOC matrices and weights for ECOC-ONE and dense random strategy.

different decision boundaries that do not approximate well the expected boundaries (fig. 3.6(d) and (e)).

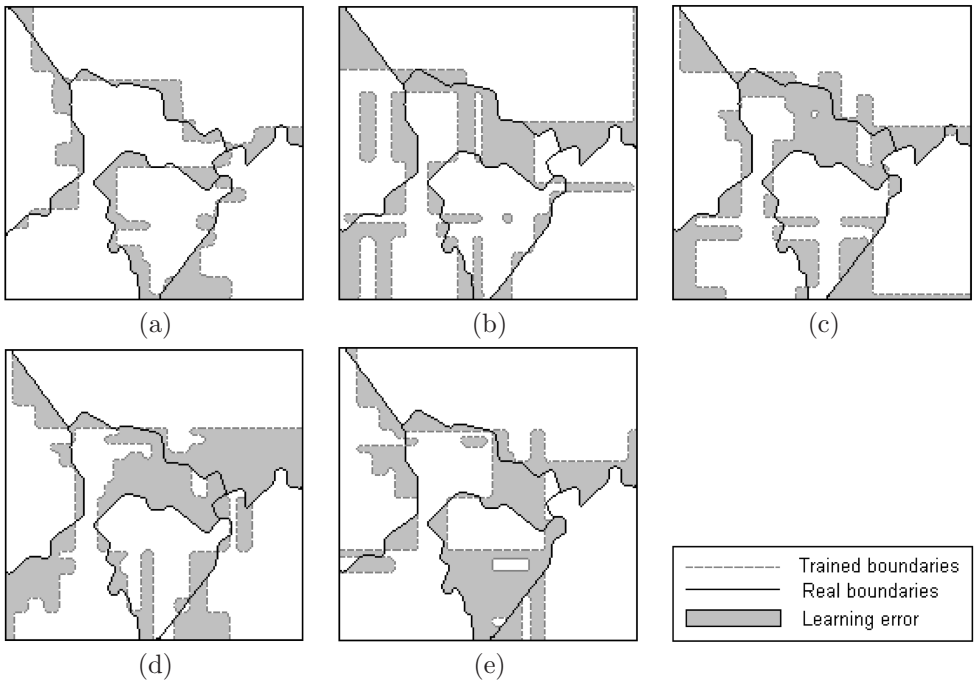
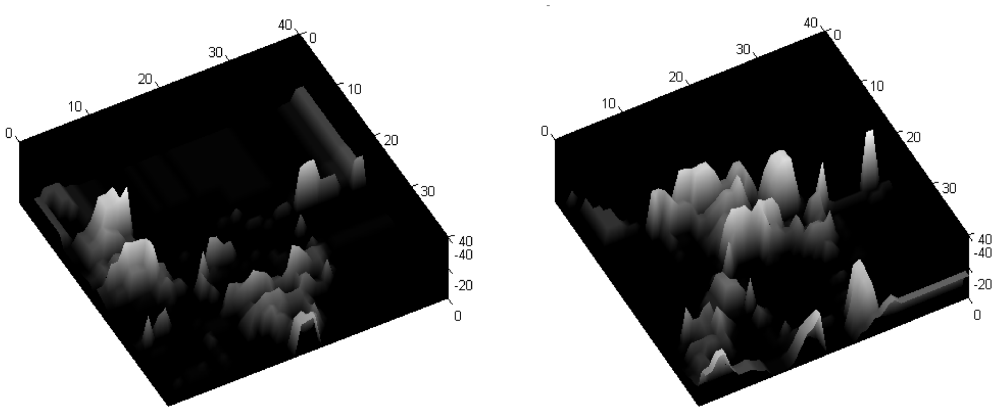


Figure 3.6: Boundaries resulted after one iteration of training. (a) ECOC-ONE, (b) one-versus-one, (c) one-versus-all and, (d) and (e) two different matrices of Dense Random with the same minimal distance, respectively. Dark line corresponds to the real boundary and grey regions correspond to learning errors.

In order to analyze the fitting of the selected dichotomizers of the ECOC-ONE matrix to the classes boundaries, the volume of the errors for the one-versus-all and ECOC-ONE technique are shown in fig. 3.7. The height corresponds to the number of times that one technique misclassifies a data sample for each spatial location. Observe that the volume of the one-versus-all technique (fig. 3.7(b)) is in this case about 70% higher than the one generated by the ECOC-ONE strategy (fig. 3.7(a)).



(a) (b)

Figure 3.7: Error surface comparison between ECOC-ONE and one-versus-all technique for the toy problem of fig. 3.3 .

3.2.6 ECOC-ONE Evaluation

In order to validate the proposed method, we use the UCI data set [8]. The data sets used are Dermathology, Ecoli, Glass, Segmentation, Vowel, Satimage, Yeast, and Pendigits (the descriptions of the UCI data sets can be found in chapter F). We compare our technique with the following ECOC coding strategies: one-versus-all ECOC (one-vs-all), one-versus-one ECOC (one-vs-one), and Dense random ECOC. The decoding process for all mentioned techniques is the standard Euclidean distance because it shows the same behavior as the Hamming decoding, but it also tends to reduce the confusion due to the use of the zero values [73]. All these strategies are compared with our ECOC-ONE method extending a tree for coding and our Attenuated Euclidean distance for decoding of section . We also include the results obtained by the ECOC-ONE computed with the MSFFS. We use a maximum of ten iterations or dichotomizers including the first optimal tree. In order to have a fair comparison, we used the same number of dichotomizers for the generation of the Dense Random ECOC matrix columns. The Dense Random matrix is selected from an exhaustive search of 10000 iterations. We have used Discriminant analysis, Discrete Adaboost with 50 decision stumps, and linear Support Vector Machines as base learners for all techniques. However, note that our technique is generic in the sense that it only uses the classification score - it is independent of the particular base classifier. All the tests are calculated using stratified ten-fold cross-validation.

Tables 3.6, 3.7 and 3.8 show the number of dichotomizers, accuracy rates and confidence intervals at 95% - we have tested for statistical significance using a two tailed t-test - for the *FLDA*, Adaboost and *SVM* techniques, respectively. The results in bold face are related to the first position in ranking of the methods which confidence interval overlaps with the one with the best accuracy - and therefore not statistically significant from the maximum mean accuracy. The rank shows the average position

of each technique. For example, if a technique obtains the best accuracy in 8 of 10 validation sets and it has been chosen as a second option in the other two sets, its rank value is 1.20. Note that all strategies with results not statistically significant from the top one are considered also as the first choice. Observing the results, we can see that our method is very competitive when compared to the other standard ECOC coding techniques. Furthermore, it attains a comparable accuracy to the one-vs-one ECOC coding strategy, which is known to usually obtain the best results. In some cases, one-vs-one improves our results for a certain data set. For example, at Pendigits data set using *FLDA*, it obtains a two percent of improvement over our method. However, one must note that one-vs-one requires 45 dichotomizers in that data set, and we use only 10. These results are easily explained by the fact that our method chooses at each step the most discriminable dichotomizer compared to the one-versus-one strategies where all pairs of classifiers are considered. Thus, our procedure allows to classify classes depending on their difficulty. For example, two difficult classes will have a high Hamming distance between rows. But two easy classes, perhaps will not have a considerable Hamming or Euclidean distance between them, since it is not necessary to correct so many errors. In this way, we can reduce the number of binary classifiers to be selected. This effect can be also seen in the results of Dense Random ECOC and our procedure. Both cases have the same number of dichotomizers (or less in our case due to the fact that we analyze the training convergence), and although Random ECOC has a higher distance between rows in most cases, our procedure usually obtains a higher hit ratio because the dichotomizers are selected in an optimal way depending on the domain of the problem. Note that the results obtained using MSFFS are usually very close to the ones obtained with the exhaustive approach. As expected, its performance is poorer than using exhaustive search. There is a trade-off between accuracy and computing time. If ECOC-ONE with exhaustive search and one-versus-one are the first choices, ECOC-ONE with MSFFS is a very close second choice. Note that there is a further trade-off in the exactitude of the MSFFS method between the optimality of the solution and the time complexity. This trade-off is governed by the number of iterations of the floating search procedure. A maximum of N iterations (where N is the number of items in the search) should suffice to obtain a good approximation [70].

Experimental discussion

In order to provide more insight on the ECOC-ONE process, we show different experiments that address the following issues: Firstly, we discuss the use of the validation sub set. Then, we show the optimality of our extension technique when it is compared with a random extension. We show an extension of the one-vs-all technique using ECOC-ONE. We compare a multi-class built-in SVM with the ECOC-ONE extension of a tree. The computational complexity of the ECOC-ONE is compared to the ECOC-ONE (MSFFS). And finally, We discuss the effect of the weights in the ECOC matrix.

In order to show the effect of the validation set, we focus on the results obtained on two data sets, the dermatology and the glass sets. Figure 3.8(a) and 3.8(b) display the error evolution using our procedure. Observe that the training error is zero in both

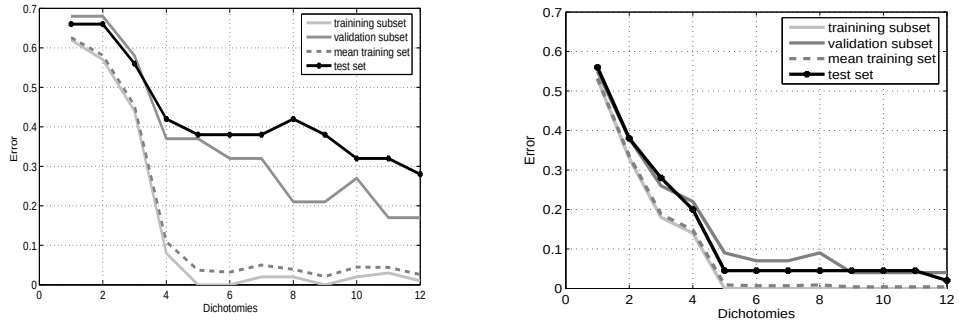


Figure 3.8: Error evolution of Dermatology data set using ECOC-ONE with FLDA. (a) error evolution for the glass data set. (b) error evolution for the dermatology data set.

cases at iteration 5. At that point further learning using the training subset is futile. However, using the validation set we still have information for accuracy improvement. In fact, looking at test evolution we can see how the test error further decreases. In general, this behavior holds even if the training error does not achieve the zero error, since the validation subset is used as an external oracle. The oracle tries to capture the variability not observed in the training set. In this way, it reinforces the learning process, serving just as an observable test.

The second experiment is designed to show the optimality of our extension technique. We increase the initial one-versus-all with the embedding of only two extra dichotomizers. Discrete Adaboost is used as a base classifier for the comparative. We compare our extension to the one-versus-all including two dense random dichotomizers (one-versus-all-dense) that maximally increase the distance between rows and columns (and its complementaries). We can observe in table 3.9 that with the reduced set of optimal extra dichotomizers, our proposed technique increases considerably the accuracy of the initial coding technique. Besides, the extension of ECOC-ONE dichotomizers seems to perform better than the extra dense dichotomizers of the comparative.

One-versus-all is considered, in general, one of the poorest choices for learning with ECOC. However, it is still used because of the small number of dichotomizers involved. The third experiment showed in this section compares the extension of the one-versus-all adding just two dichotomizers using our method with the one-versus-one approach - recall that one-versus-one is the standard technique with highest accuracy. In order to perform this comparison we have used Discrete Adaboost on the UCI repository. Table 3.10 shows the results of these experiments. Observe that both methods achieve the same performance considering the confidence interval at 95%. Note also that the number of dichotomizers involved in our extension is smaller than the one-versus-one approach.

In order to further validate our approach, we provide a new experiment comparing the ECOC-ONE technique using Support Vector Machines with linear kernels with a built-in multi-class SVM [37] with the same kernel. The results are shown in table

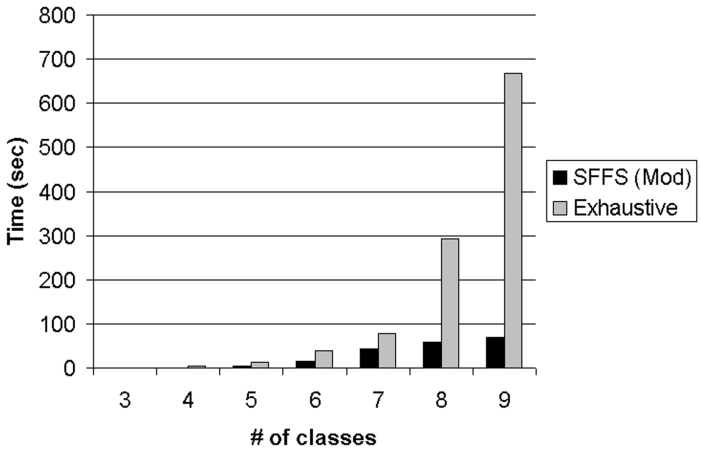


Figure 3.9: Time consumed by the exhaustive search and MSFFS.

3.11. Observe that our technique slightly improves the accuracy of the multi-class SVM using the same parametrization for both techniques.

In order to reduce the computational complexity of the exhaustive search when the number of classes is high, we propose the use of the modified sequential forward floating search (MSFFS). We have designed an experiment that shows the difference in complexity between the MSFFS and the exhaustive search. Using the Pendigits data set, we compute the time of finding a sub-optimal column of the ECOC matrix as the number of classes increases.

Figure 3.9 illustrates the results of the experiment. Observe the exponential behavior of the exhaustive search and the quasi-linear tendency of the MSFFS. As we have shown in the former section, the results using this sub-optimal search technique are very similar to those obtained using the exhaustive search.

As commented in former sections, the dichotomizers are selected in an optimal way in order to ensure generalization of the proposed approach. Each of the selected dichotomizers corrects a certain partition of the subset of classes and has associated an error according to the training and validation subset of misclassified samples. We use the classification score to weight each dichotomizer using the empiric error of classification for that dichotomizer using eq.(3.2). Figure 3.10 shows the average and relative improvement of the weighted Euclidean distance referred to the error obtained using just the Euclidean distance. Besides, we present the figures that reflect the effect of the weighted distance (table 3.12). The results show that the weighting scheme increases the accuracy in all cases, showing the absolute and relative improving percentages. Besides, we can observe that the variance is clearly reduced by the fact that in all cases - except for the Ecoli data-set - the confidence rate is smaller.

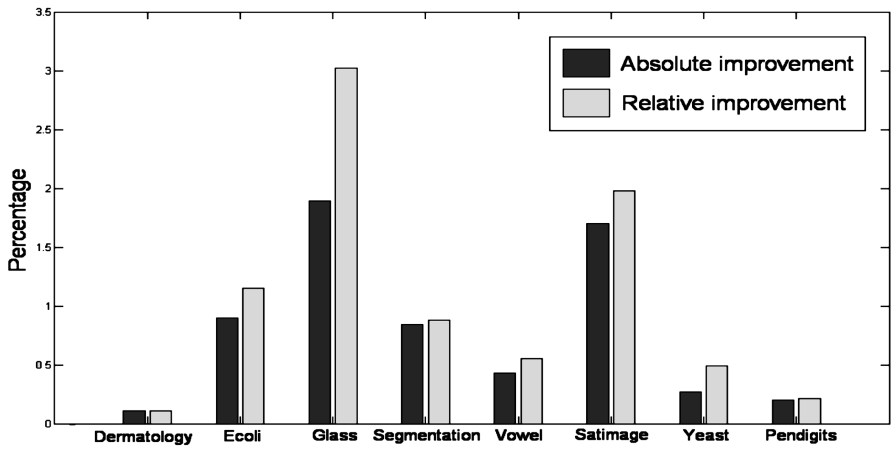


Figure 3.10: Absolute and relative percentage improvement comparison between Euclidean distance and weighted Euclidean distance

Table 3.4: Modified sequential forward floating search algorithm

Input:

$Y = \{y_j | j = 1..D\}$//Pool of available items//

Output:

$X_k^1 = \{x_l | l = 1..|Y|(orD), x_l \in Y\}; \quad X_k^2 = \{x_m | m = 1..|Y|, x_m \in (Y/X_k^1)\}$

Initialization:

$X_0^1 = X_0^2 = \{\emptyset\}; \quad k = 0$

Termination:

Stop when $|J(X_k^1, X_k^2) - J(X_{k-1}^1, X_{k-1}^2)| \leq \epsilon$

Step 1 (Inclusion)

$x_+^{'} = argmax_{x \in Y/\{X_k^1 \cup X_k^2\}} \; J(X_k^1 \cup x, X_k^2);$
$x_+^{''} = argmax_{x \in Y/\{X_k^1 \cup X_k^2\}} \; J(X_k^1, X_k^2 \cup x)$

$$(X_{k+1}^1, X_{k+1}^2) = \begin{cases} (X_k^1 \cup x_+^{'}, X_k^2) & \text{if } J(X_k^1 \cup x_+^{'}, X_k^2) > J(X_k^1, X_k^2 \cup x_+^{''}) \\ (X_k^1, X_k^2 \cup x_+^{''}) & \text{if } J(X_k^1 \cup x_+^{'}, X_k^2) < J(X_k^1, X_k^2 \cup x_+^{''}) \end{cases}$$

$k = k + 1$

Step 2 (Conditional exclusion)

$x_-^{'} = argmax_{x \in Y/X_k^1} \; J(X_k^1/x, X_k^2); \quad x_-^{''} = argmax_{x \in Y/X_k^2} \; J(X_k^1, X_k^2/x)$

$$(X_{k+1}^1, X_{k+1}^2) = \begin{cases} (X_k^1/x_-^{'}, X_k^2) & \text{if } J(X_k^1/x_-^{'}, X_k^2) > J(X_k^1, X_k^2) \text{ and} \\ & J(X_k^1/x_-^{'}, X_k^2) > J(X_k^1, X_k^2/x_-^{''}) \\ (X_k^1, X_k^2/x_-^{''}) & \text{if} J(X_k^1, X_k^2/x_-^{''}) > J(X_k^1, X_k^2) \text{ and} \\ & J(X_k^1/x_-^{'}, X_k^2) < J(X_k^1, X_k^2/x_-^{''}) \end{cases}$$

$k = k + 1$

if $J(X_k^1, X_k^2/x_-^{''}) > J(X_k^1, X_k^2)$ **or** $J(X_k^1/x_-^{'}, X_k^2) > J(X_k^1, X_k^2)$
then go to **Step 2**
else go to **Step 1**

Table 3.5: ECOC strategies hits for a toy problem (#D means number of dichotomizers).

one-vs-one ECOC		one-vs-all ECOC		Dense random ECOC		ECOC-ONE	
Hit	#D	Hit	#D	Hit	#D	Hit	#D
70.83±1.17	6	66.67±1.07	4	67.67±1.91	5	**72.92±0.82**	5

Table 3.6: ECOC Strategies hits for UCI data sets using *FLDA* as a base classifier.

#	one-vs-one		one-vs-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	96.65±0.73	15	94.87±0.74	6	96.57±0.74	10	**98.48±0.49**	7.8	96.03±0.97	10
(b)	**82.40±1.46**	28	71.85±1.53	8	**81.15±1.55**	10	**83.90±1.23**	10	**81.73±2.14**	10
(c)	**76.76±1.16**	21	44.55±2.15	7	44.83±2.00	10	52.10±2.28	10	51.65±1.87	10
(d)	**85.24±0.57**	21	71.32±0.62	7	73.92±0.56	10	**85.44±0.50**	9.2	**84.65±1.05**	10
(e)	**71.20±1.27**	55	23.87±0.42	11	41.32±1.38	10	53.05±0.80	10	51.04±1.42	10
(f)	81.00±0.67	15	65.35±0.52	6	75.85±0.83	10	**82.85±0.54**	9.4	80.48±0.85	10
(g)	**52.21±0.80**	45	30.54±0.90	10	47.32±0.93	10	**51.21±0.70**	10	**50.67±1.35**	10
(h)	**93.18±0.43**	45	33.10±1.23	10	68.41±1.44	10	**91.21±0.78**	10	**91.03±1.23**	10
Rank	**1.25**		3.87		2.75		**1.25**		2.12	

Table 3.7: ECOC Strategies hits for UCI data sets using Discrete Adaboost as a base classifier.

#	one-vs-one		one-vs-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	**96.30±0.61**	15	92.65±1.23	6	**95.26±0.82**	10	**95.17±0.74**	8.2	**95.11±0.71**	10
(b)	**78.05±1.46**	28	**77.10±1.19**	8	**77.65±1.33**	10	**78.15±1.84**	10	**77.14±1.55**	10
(c)	**67.93±1.66**	21	60.83±2.34	7	63.69±2.51	10	**67.03±1.63**	10	**66.55±1.76**	10
(d)	**97.01±0.72**	21	92.89±1.16	7	94.51±1.22	10	**96.23±1.52**	9.6	94.38±1.84	10
(e)	**81.43±1.12**	55	73.33±1.40	11	74.50±1.96	10	**81.50±1.22**	10	**80.83±2.53**	10
(f)	**86.23±0.79**	15	81.99±0.86	6	84.39±0.76	10	**85.47±1.00**	9.8	**84.67±2.17**	10
(g)	**52.35±1.05**	45	**51.48±1.08**	10	**51.82±1.47**	10	**52.87±1.96**	10	**52.87±1.96**	10
(h)	**98.01±1.04**	45	93.98±2.56	10	**95.54±1.71**	10	**97.84±1.13**	10	**97.09±1.56**	10
Rank	**1.00**		2.37		1.50		**1.00**		**1.25**	

Table 3.8: ECOC Strategies hits for UCI data sets using Linear *SVM* as a base classifier.

#	one-vs-one		one-vs-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	**96.02±0.95**	15	**94.83±1.84**	6	**95.94±1.22**	10	**95.83±0.94**	8.7	**95.72±1.01**	10
(b)	**76.11±1.26**	28	63.97±1.51	8	72.94±1.37	10	**75.68±1.28**	10	**74.75±1.48**	10
(c)	**58.52±2.63**	21	49.73±2.45	7	54.13±2.73	10	**57.83±1.93**	10	**56.79±1.21**	10
(d)	**98.36±1.47**	21	94.36±1.13	7	93.83±1.43	10	**97.84±1.12**	9.2	**96.84±1.52**	10
(e)	**73.18±1.15**	55	32.07±1.62	11	46.00±1.34	10	**69.14±3.01**	10	67.65±4.02	10
(f)	87.43±0.80	15	85.85±1.08	6	84.03±1.49	10	**89.04±0.63**	10	**88.01±0.97**	10
(g)	**55.31±1.47**	45	41.41±1.79	10	51.07±2.12	10	**52.58±1.73**	10	**52.49±2.13**	10
(h)	**98.53±1.03**	45	95.04±1.88	10	**96.44±1.12**	10	**98.43±0.99**	10	**96.05±1.76**	10
Rank	1.13		2.62		2.25		**1.00**		1.13	

Table 3.9: UCI one-vs-all extension using Discrete Adaboost.

Problem	one-vs-all ECOC	one-vs-all-dense ECOC	one-vs-all-ONE ECOC
Dermathology	92.65±1.23	93.85±1.02	**95.53±0.89**
Ecoli	**77.10±1.19**	**77.58±1.54**	**78.43±1.02**
Glass	60.83±2.34	**65.59±2.52**	**64.90±2.39**
Segmentation	92.89±1.16	**94.80±1.21**	**95.90±1.03**
Vowel	73.33±1.40	74.97±1.40	**79.34±1.40**
Satimage	81.99±0.86	**83.93±1.11**	**84.83±0.96**
Yeast	51.48±1.08	51.48±1.08	**53.52±0.89**
Pendigits	**93.98±2.56**	**95.64±1.89**	**96.88±2.01**
Rank	2.50	1.38	**1.00**

Table 3.10: UCI one-versus-one and one-versus-all-ONE ECOCs comparison.

Problem	one-vs-one ECOC		one-vs-all ECOC-ONE	
	Hit	#D	Hit	#D
Dermathology	**96.30±0.61**	15	**95.53±0.89**	8
Ecoli	**78.05±1.46**	28	**78.43±1.02**	10
Glass	**67.93±1.66**	21	**64.90±2.39**	9
Segmentation	**97.01±0.72**	21	**95.90±1.03**	9
Vowel	**81.43±1.12**	55	**79.34±1.40**	13
Satimage	**86.23±0.79**	15	**84.83±0.96**	8
Yeast	**52.35±1.05**	45	**53.52±0.89**	12
Pendigits	**98.01±1.04**	45	**96.88±2.01**	12

Table 3.11: UCI ECOC-ONE with SVM and built-in multi-class SVM with lineal kernel comparative.

Problem	ECOC-ONE	Multiclass SVM
Dermathology	**95.83±0.94**	**96.52±0.61**
Ecoli	**75.68±1.28**	69.74±0.76
Glass	**57.83±1.93**	**59.93±1.99**
Segmentation	**97.84±1.12**	95.23±0.59
Vowel	69.14±3.01	**77.55±0.96**
Satimage	**89.04±0.63**	85.60±0.40
Yeast	**52.58±1.73**	**52.57±0.92**
Pendigits	**98.43±0.99**	**98.72±0.17**
Rank	**1.12**	1.38

Table 3.12: Accuracy of the Euclidean and weighted Euclidean decoding at UCI data sets using Discrete Adaboost and $N \times 2$ columns, being N the number of classes, and dense random coding.

	Dermathology	Ecoli	Glass	Segmentation
Euclidean	96.74±0.79	78.39±1.43	62.59±2.74	95.38±1.51
Weighted	**96.85±0.73**	**79.29±1.53**	**64.48±2.60**	**96.22±1.20**
% Absolute	+0.11	+0.90	+1.89	+0.84
% Relative	+0.11	+1.15	+3.02	+0.88
	Vowel	Satimage	Yeast	Pendigits
Euclidean	78.10±2.38	85.80±1.49	54.73±1.66	96.95±1.05
Weighted	**78.53±2.32**	**87.50±1.03**	**55.00±1.46**	**97.15±0.95**
% Absolute	+0.43	+1.70	+0.27	+0.20
% Relative	+0.55	+1.98	+0.49	+0.21

3.3 Sub-class ECOC

One of the main reasons why the present problem-dependent designs attain a good performance is because of the high number of possible sub-groups of classes that is possible in the ternary ECOC framework. On the other hand, considering the training data in the process of the ECOC design allows to obtain compact codewords with high classification performance. However, the final accuracy is still based on the ability of the base classifier to learn each individual problem. Difficult problems, those which the base classifier is not able to find a solution for, require the use of complex classifiers, such as Support Vector Machines with Radial Basis Function kernel [37], and expensive parameter optimizations. Look at the example of fig. 3.11(a). A linear classifier is used to split two classes. In this case, the base classifier is not able to find a convex solution. On the other hand, in fig. 3.11(b), one of the previous classes has been split into two sub-sets, that we call *sub-classes*. Then, the original problem is solved using two linear classifiers, and the two new sub-classes have the same original class label. Some studies in the literature tried to form sub-classes using the labels information, which is called Supervised Clustering [96][21]. In these types of systems, clusters are usually formed without taking into account the behavior of the base classifier that learns the data. In a recent work [102], the authors use the class labels to form the sub-classes that improve the performance of particular Discriminant Analysis algorithms.

In this section, we present a problem-dependent ECOC design where classes are partitioned into sub-classes using a clustering approach for the cases that the base classifier is not capable to distinguish the classes. Sequential Forward Floating Search based on maximizing the Mutual Information is used to generate the sub-groups of problems that are split into more simple ones until the base classifier is able to learn the original problem. In this way, multi-class problems which can not be modeled by using the original set of classes are modeled without the need of using more complex classifiers. The final ECOC design is obtained by combining the sub-problems.

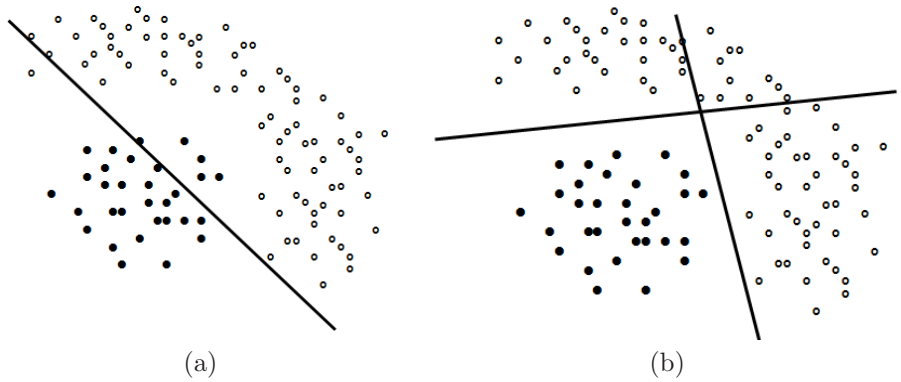


Figure 3.11: (a) Decision boundary of a linear classifier of a 2-class problem. (b) Decision boundaries of a linear classifier splitting the problem of (a) into two more simple tasks.

3.3.1 Problem-dependent ECOC Sub-class

From an initial set of classes C of a given multi-class problem, the objective of the Sub-class ECOC strategy is to define a new set of classes C', where $|C'| > |C|$, so that the new set of binary problems is easier to learn for a given base classifier. For this purpose, we use a guided procedure that, in a problem-dependent way, groups classes and splits them into sub-sets if necessary.

Look at the 3-class problem shown on the top of fig. 3.12(a). The standard DECOC algorithm [73] considers the whole set of classes to split it into two sub-sets of classes $\wp^+$ and $\wp^-$ maximizing the *MI* criterion on a sequential forward floating search procedure (*SFFS*). In the example, the first sub-sets found correspond to $\wp^+ = \{C_1, C_2\}$ and $\wp^- = \{C_3\}$. Then, a base classifier is used to train its corresponding dichotomizer h_1. This classifier is shown in the node h_1 of the tree structure shown in fig. 3.12(d). The procedure is repeated until all classes are split into separate sub-sets $\wp$. In the example, the second classifier is trained to split the sub-sets of classes $\wp^+ = C_1$ from $\wp^- = C_2$ because the classes C_1 and C_2 were still contained in a single sub-set after the first step. This second classifier is codified by the node h_2 of fig. 3.12(d). When the tree is constructed, the coding matrix M is obtained by codifying each internal node of the tree as a column of the coding matrix (see fig. 3.12(c)).

In our case, sequential forward floating search (*SFFS*) is also applied to look for the sub-sets $\wp^+$ and $\wp^-$ that maximizes the mutual information between the data and their respective class labels [73]. The encoding algorithm is shown in table 3.13.

Given a N-class problem, the whole set of classes is used to initialize the set L containing the sets of labels for the classes to be learned. At the beginning of each iteration k of the algorithm (**Step 1**), the first element of L is assigned to S_k in the first step of the algorithm. Next, $SFFS$ is used to find the optimal binary partition BP of S_k that maximizes the mutual information I between the data and their respective class labels (**Step 2**). The *SFFS* algorithm used [70] is shown in Appendix I, and the implementation details of the fast quadratic mutual information can be found in Appendix II.

To illustrate our procedure, let us return to the example of the top of fig. 3.12(a). On the first iteration of the sub-class ECOC algorithm, $SFFS$ finds the sub-set $\wp^+ = \{C_1, C_2\}$ against $\wp^- = \{C_3\}$. The encoding of this problem is shown in the first matrix of fig. 3.12(c). The positions of the column corresponding to the classes of the first partition are coded by +1 and the classes corresponding to the second partition to -1, respectively. In our procedure, the base classifier is used to test if the performance obtained by the trained dichotomizers is sufficient. Observe the decision boundaries of the picture next to the first column of the matrix in fig. 3.12(b). One can see that the base classifier finds a good solution for this first problem.

Then, the second classifier is trained to split $\wp^+ = C_1$ against $\wp^- = C_2$, and its performance is computed. To separate the current sub-sets is not a trivial problem, and the classification performance is poor. Therefore, our procedure tries to split the data $J_{\wp^+}$ and $J_{\wp^-}$ from the current sub-sets $\wp^+$ and $\wp^-$ into more simple sub-sets. At **Step 3** of the algorithm, the splitting criteria SC takes as input a data set $J_{\wp^+}$ or $J_{\wp^-}$ from a sub-set $\wp^+$ or $\wp^-$, and splits it into two sub-sets $J^+_{\wp^+}$ and $J^-_{\wp^+}$ or $J^+_{\wp^-}$ and $J^-_{\wp^-}$. On the experimental results chapter we discuss the selection of the splitting

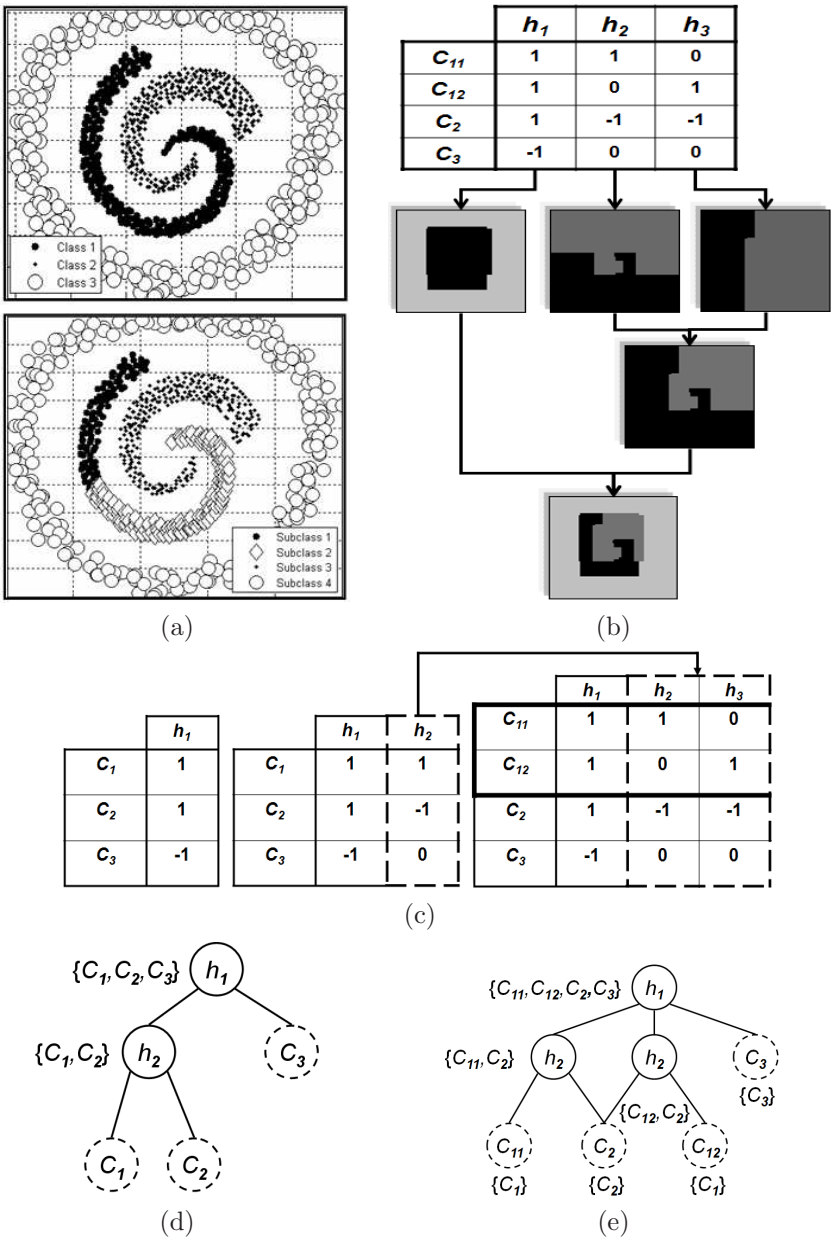


Figure 3.12: (a) Top: Original 3-class problem. Bottom: 4 sub-classes found. (b) Sub-class ECOC encoding using the four sub-classes using Discrete Adaboost with 40 runs of Decision Stumps. (c) Learning evolution of the sub-class matrix M. (d) Original tree structure without applying sub-class. (e) New tree-based configuration using sub-classes.

Table 3.13: Problem-dependent Sub-class ECOC algorithm.

Inputs: $J, C, \theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$ //Thresholds for the number of samples, performance, and improvement between iterations

Outputs: $C', J', \wp', M$

[**Initialization:**]
Create the trivial partition $\{\wp_0^+, \wp_0^-\}$ of the set of classes $\{C_i\}$:
$\{\wp_0^+, \wp_0^-\} = \{\{\emptyset\}, \{C_1, C_2, ..., C_N\}\}$
$L_0 = \{\wp_0^-\}; J' = J; C' = C; \wp' = \emptyset; M = \emptyset; k = 1$

Step 1 S_k is the first element of L_{k-1}
$L'_k = L_{k-1} \backslash \{S_k\}$

Step 2 Find the optimal binary partition $BP(S_k)$:
$\{\wp_k^+, \wp_k^-\} = argmax_{BP(S_k)}(I(\mathbf{x}, d(BP(S_k))))$
where I is the mutual information criterion, $\mathbf{x}$ is the random variable associated to the features and d is the discrete random variable of the dichotomizer labels[a], defined in the following terms,

$$d = d(\mathbf{x}, BP(S_k)) = \begin{cases} 1 & \text{if } \mathbf{x} \in C_i | C_i \in \wp_k^+ \\ -1 & \text{if } \mathbf{x} \in C_i | C_i \in \wp_k^- \end{cases}$$

Step 3 // Look for sub-classes
$\{C', J', \wp'\} = SPLIT(J_{p_k^+}, J_{p_k^-}, C', J', J, \wp', \theta)^b$

Step 4 $L_k = \{L'_k \cup \wp_k^i\}$ if $|\wp_k^i| > 1 \ \forall i \in \{+, -\}$

Step 5 If $|L_k| \neq 0$
$k = k + 1$ **go to Step 1**

Step 6 Codify the coding matrix M using each partition $\{\wp_i^+, \wp_i^-\}$ of $\wp', i \in [1, .., |\wp'|]$ and each class $C_r \in \wp_i = \{\wp_i^+ \cup \wp_i^-\}$ as follows:

$$M(C_r, i) = \begin{cases} 0 & \text{if } C_r \notin \wp_i \\ +1 & \text{if } C_r \in \wp_i^+ \\ -1 & \text{if } C_r \in \wp_i^- \end{cases} \tag{3.7}$$

[a] Use $SFFS$ of Appendix I as the maximization procedure and MI of Appendix II to estimate I

[b] Using the splitting algorithm of table 3.14.

criterion. The splitting algorithm is shown in table 3.14.

When two data sub-sets $\{J_{\wp^+}^+, J_{\wp^+}^-\}$ and $\{J_{\wp^-}^+, J_{\wp^-}^-\}$ are obtained, only one of both split sub-sets is used. We select the sub-sets that have the highest distance between the means of each cluster. Suppose that the distance between $J_{\wp^-}^+$ and $J_{\wp^-}^-$ is larger than between $J_{\wp^+}^+$ and $J_{\wp^+}^-$. Then, only $J_{\wp^+}$, $J_{\wp^-}^+$, and $J_{\wp^-}^-$ are used. If the new

Table 3.14: Sub-class *SPLIT* algorithm.

Inputs: $J_{\wp^1}, J_{\wp^2}, C', J', J, \wp', \theta$ // C' is the final set of classes, J' the data for the final set of classes, and $\wp'$ is the labels for all the partitions of classes of the final set.

Outputs: $C', J', \wp'$

Step 1 Split problems:

$\{J^+_{\wp^+}, J^-_{\wp^+}\} = SC(J_{\wp^+})^a$

$\{J^+_{\wp^-}, J^-_{\wp^-}\} = SC(J_{\wp^-})$

Step 2 Select sub-classes:

if $|\overline{J^+_{\wp^+}}, \overline{J^-_{\wp^+}}| > |\overline{J^+_{\wp^-}}, \overline{J^-_{\wp^-}}|$ // find the largest distance between the means of each sub-set.

$\{J^+_+, J^-_+\} = \{J^+_{\wp^+}, J_{\wp^-}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^+}, J_{\wp^-}\}$

else

$\{J^+_+, J^-_+\} = \{J^+_{\wp^-}, J_{\wp^+}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^-}, J_{\wp^+}\}$

end

Step 3 Test parameters to continue splitting:

if $TEST\_PARAMETERS(J_{\wp^1}, J_{\wp^2}, J^1_1, J^2_1, J^1_2, J^2_2, \theta)$// call the function with the new sub-sets

$\{C', J', \wp'\} = SPLIT(J^1_1, J^2_1, C', J', J, \wp', \theta)$

$\{C', J', \wp'\} = SPLIT(J^1_2, J^2_2, C', J', J, \wp', \theta)$

end

Step 4 Save the current partition:

Update the data for the new sub-classes and previous sub-classes if intersections exists J'.

Update the final number of sub-classes C'.

Create $\wp_c = \{\wp_{c^1}, \wp_{c^2}\}$ the set of labels of the current partition.

Update the labels of the previous partitions $\wp$.

Update the set of partitions labels with the new partition $\wp' = \wp' \cup \wp_c$.

[a] SC corresponds to the splitting method of the input data into two main clusters.

sub-sets improve the classification performance, new sub-classes are formed, and the process is repeated.

In the example of fig. 3.12, applying the splitting criteria SC over the two sub-sets, two clusters are found for $\wp^+ = C_1$ and for $\wp^- = C_2$. Then, the original encoding of the problem C_1 vs C_2 (corresponding to the second column of the matrix in the

center of fig. 3.12(c)) is split into two columns marked with the dashed lines in the matrix on the right. In this way, the original C_1 vs C_2 problem is transformed to two more simple problems $\{C_{11}\}$ against $\{C_2\}$ and $\{C_{12}\}$ against $\{C_2\}$. Here the first subindex of the class corresponds to the original class, and the second subindex to the number of sub-class. It implies that the class C_1 is split into two sub-classes (look at the bottom of fig. 3.12(a)), and the original 3-class problem $C = \{C_1, C_2, C_3\}$ becomes the 4-sub-class problem $C' = \{C_{11}, C_{12}, C_2, C_3\}$. As the class C_1 has been decomposed by the splitting of the second problem, we need to save the information of the current sub-sets and the previous sub-sets affected by the new splitting. The steps to update this information are summarized in the **Step 4** of the splitting algorithm. We use the object labels to define the set of sub-classes of the current partition $\wp_c$. If new sub-classes are created, the set of sub-classes C' and the data for sub-classes J' have to be updated. Note that when a class or a sub-class previously considered for a given binary problem is split in a future iteration of the procedure, the labels from the previous sub-sets $\{\wp^+, \wp^-\}$ need to be updated with the new information. Finally, the set of labels for the binary problems $\wp'$ is updated with the labels of the current sub-set $\wp' = \wp' \cup \wp_c$. In the example of fig. 3.12, the dichotomizer h_1 considers the sub-sets $\wp_1^+ = \{C_1, C_2\}$ and $\wp_1^- = \{C_3\}$. Then, those positions containing class C_1 are replaced with C_{11} and C_{12}. The process is repeated until the desired performance is achieved or the stopping conditions are full-filled.

The conditions that guide the learning and splitting process are defined by the set of parameters $\theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$, where θ_{size} corresponds to the minimum size of a sub-set to be clustered, θ_{perf} contains the minimum error desired for each binary problem, and θ_{impr} looks for the improvement of the split sub-sets regarding the previous ones. The function $TEST\_PARAMETERS$ in table 3.14 is responsible for testing the constraints based on the parameters $\{\theta_{size}, \theta_{perf}, \theta_{impr}\}$. If the constraints are satisfied, the new sub-sets are selected and used to recursively call the splitting function (**Step 3** of the algorithm in table 3.14). The constraints of the function $TEST\_PARAMETERS$ are fixed by default as follows:

- The number of objects in $J_{\wp^+}$ has to be larger than θ_{size}.
- The number of objects in $J_{\wp^-}$ has to be larger than θ_{size}.
- The error $\xi(h(J_{\wp^-}, J_{\wp^+}))$ obtained from the dichomomizer h using a particular base classifier applied on the sets $\{\wp^+, \wp^-\}$ has to be larger than θ_{perf}.
- The sum of the well-classified objects from the two new problems (based on the confusion matrices) divided by the total number of objects has to be greater than $1 - \theta_{impr}$.

θ_{size} avoids the learning of very unbalanced problems. θ_{perf} determines when the performance of a partition of classes is insufficient and sub-classes are required. And finally, when a partition does not obtain the desired performance θ_{perf}, the splitting of the data stops, preventing overtraining.

In the example of fig. 3.12, the three dichotomizers h_1, h_2, and h_3 find a solution for the problem (look the trained boundaries shown in fig. 3.12(b)), obtaining a classification error under θ_{perf}, so, the process stops. Now, the original tree encoding of the DECOC design shown in fig. 3.12(d) can be represented by the tree structure of fig. 3.12(e), where the original class associated to each sub-class is shown in the leaves.

Summarizing, when a set of objects belonging to different classes is split, object labels are not taken into account. It can be seen as a clustering in the sense that the sub-sets are split into more simple ones while the splitting constraints are satisfied. It is important to note that when one uses different base classifiers, the sub-class splitting is probably applied to different classes or sub-classes, and therefore, the final number of sub-classes and binary problems differs.

When the final set of binary problems is obtained, its respective set of labels $\wp'$ is used to create the coding matrix M (eq. (3.7)). The outputs C' and J' contain the final set of sub-classes and the new data for each sub-class, respectively.

Finally, to decode the new sub-class problem-dependent design of ECOC, we take advantage of the Loss-Weighted decoding design of section 4.5. The decoding strategy uses a set of normalized probabilities based on the performance of the base classifier and the ternary ECOC constraints.

Illustration over toy problems

To show the effect of the Sub-class ECOC strategy for different base classifiers, we used the previous toy problem of the top of fig. 3.12(a). Five different base classifiers are applied: Fisher Linear Discriminant Analysis (*FLDA*), Discrete Adaboost, Nearest Mean Classifier, Linear *SVM*, and *SVM* with Radial Basis Function kernel[7]. Using these base classifiers on the toy problem, the original DECOC strategy with the Loss-Weighted algorithm obtains the decision boundaries shown on the top row of fig. 3.13. The new learned boundaries are shown on the bottom row of fig. 3.13 for fixed parameters θ. Depending on the flexibility of the base classifier more sub-classes are required, and thus, more binary problems. Observe that all base classifiers are able to find a solution for the problem, although with different types of decision boundaries.

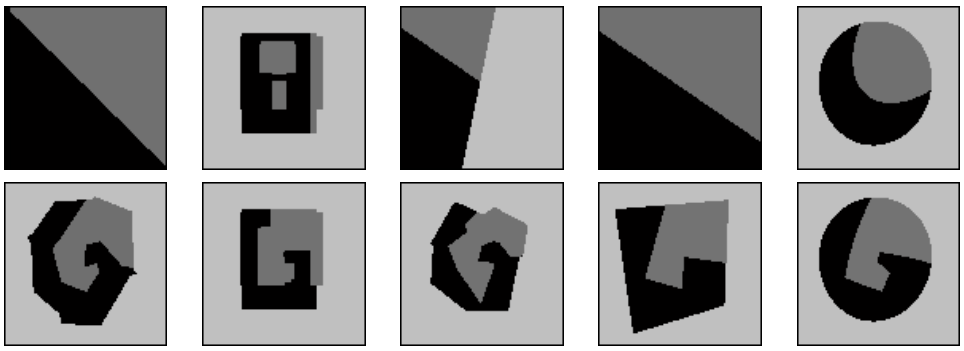


Figure 3.13: Sub-class ECOC without sub-classes (top) and including sub-classes (bottom): for *FLDA* (a), Discrete Adaboost (b), *NMC* (c), Linear *SVM* (d), and *RBF SVM* (e).

The selection of the set of parameters θ has a decisive influence on the final results.

[7]The parameters of the base classifiers are explained in the evaluation section.

We can decrease the value of θ_{perf} and increase the value of θ_{impr} to obtain a better solution for a problem, but we need to optimize the parameters to avoid overtraining by stopping the procedure if no more improvement can be achieved. In the same way, sometimes to obtain the best solution for a problem implies to learn more simple problems. These points should be considered to obtain the desired trade-off between performance and computational cost. A simple example to show the evolution of learning for different parameters θ over the previous problem is shown in fig. 3.14. The base classifier applied is $FLDA$. One can observe that when θ_{perf} decreases, more dichotomizers are required to obtain a higher performance. Thus, to achieve the desired accuracy, more sub-classes and binary problems are needed.

(a) (b) (c) (d) (e)

Figure 3.14: Learned boundaries using $FLDA$ with $\theta_{size} = \frac{|J|}{50}$, $\theta_{impr} = 0.95$, and $\theta_{perf} = 0.2$ (a), $\theta_{perf} = 0.15$ (b), $\theta_{perf} = 0.1$ (c), $\theta_{perf} = 0.05$ (d), and $\theta_{perf} = 0$ (e), respectively.

3.3.2 Sub-class ECOC Evaluation

In this section, we compare the Sub-class approach with different state-of-the-art coding designs and base classifiers on different data sets. In order to evaluate the methodology, first we discuss the data, compared methods, experiments, and performance evaluation[8].

• *Data*: The data used for the experiments consists of eight arbitrary multi-class data sets from the UCI Machine Learning Repository [8]: Iris, Ecoli, Wine, Glass, Thyroid, Vowel, Balance, and Yeast (the details of the UCI data sets can be found in chapter F). All data sets have been normalized with respect to the mean and variance.

• *Compared methods*: We compare our method with the state-of-the-art ECOC coding designs: one-versus-one [88], one-versus-all [68], dense random [5], sparse random [5], and DECOC [73].

The random matrices were selected from a set of 20000 randomly generated matrices, with $P(1) = P(-1) = 0.5$ for the dense random matrix and $P(1) = P(-1) = P(0) = 1/3$ for the sparse random matrix. The number of binary problems was fixed to the number of classes. Therefore, a direct comparison to the one-versus-all and DECOC designs is possible. Each strategy uses the previously mentioned Linear Loss-weighted decoding to evaluate their performances at identical conditions. Five different base classifiers are applied over each ECOC configuration: Nearest

[8]More experimental results and analysis of the Sub-class ECOC methodology are shown in chapter 7.

Mean Classifier (*NMC*) with the classification decision using the Euclidean distance between the mean of the classes, Discrete Adaboost with 40 iterations of Decision Stumps [32], Linear Discriminant Analysis implementation of the PR Tools using the default values [3], OSU implementation of Linear Support Vector Machines with the regularization parameter C set to 1 [37], and OSU implementation of Support Vector Machines with Radial Basis Function kernel with the default values of the regularization parameter C and the gamma parameter set to 1 [66][9].

• *Experiments*: First, we classify the set of UCI Machine Learning Repository data sets with the ECOC designs and the different base classifiers. Second, focusing on particular data sets, we analyze the performance of our methodology over the training and test sets by changing the values of the set of parameters θ. Moreover, we also perform an experiment to show the behavior of our procedure when working with different training sizes.

• *Performance evaluation*: To evaluate the performance of the different experiments, we apply stratified ten-fold cross-validation and test for the confidence interval at 95% with a two-tailed t-test. Moreover, we use the statistical Nemenyi test to look for significant differences between the method performances [22].

UCI Machine Learning Repository

Using the UCI Machine Learning Repository data sets, we perform different experiments. First, we classify the eight data sets. Second, we look for the statistical significance of the results, and final, we discuss the effect of the sub-class parameters and the training size.

UCI Machine Learning Repository classification

Using the previous eight UCI data sets, the five base classifiers, and the six ECOC designs, we have performed a total of 240 ten-fold tests. The set of parameters of the sub-class approach $\theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$ has been fixed to $\theta_{size} = \frac{|J|}{50}$ minimum number of objects to apply sub-class (thus, 2% of the samples of each particular problem), $\theta_{perf} = 0$ to split classes if the binary problem does not learn properly the training objects, and $\theta_{impr} = 0.95$, that means that the split problems must improve at least a 5% of the performance of the problem without splitting. The last measure is simply estimated by dividing the sum of the well-classified objects from the two sub-problems the total number of objects by looking at the confusion matrices. For simplicity and fast computation, the used splitting criterion is k-means with k=2. k-means is a fast way to split a set of objects into k clusters satisfying intra-cluster compactness and high inter-cluster separability by minimizing an objective function:

$$K_m = \sum_{i=1}^{2} \sum_{j=1}^{m} \parallel x_i^j - \zeta_j \parallel^2 \tag{3.8}$$

[9]For all the experiments of this chapter and the rest of the chapters of the thesis, the regularization parameter C and the gamma parameter are set to 1 for Linear and Radial Basis function Support Vector Machines. We selected this parameter after a preliminary set of experiments. We decided to keep the parameter fixed for the sake of simplicity and easiness of replication of the experiments, though we are aware that this parameter might not be optimal for all data sets. However, since the parameters are the same for all the compared methods, any weakness in the results will also be shared.

for m object instances, ζ_j the centroid for the cluster $i \in \{1, 2\}$, and K_m the objective function to be minimized[10].

The results for each base classifier are shown in the tables 3.17 to 3.19. Each position of the table contains the performance obtained applying ten-fold cross-validation and the confidence interval at 95%. The mean number of classes (or sub-classes) and the mean number of binary problems are shown below each performance. In fig. 3.15 and 3.16, the results are graphically illustrated for the Discrete Adaboost and *NMC* classifiers, respectively.

Table 3.15: UCI repository experiments for Discrete Adaboost.

Problem	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Balance	78.3(4.9) 3×3	76.5(5.9) 3×3	80.9(6.2) 3×3	78.8(4.1) 3×3	78.3(4.9) 3×2	79.0(5.0) 3.8×3.2
Wine	93.7(1.3) 3×3	96.05(1.2) 3×3	96.0(1.2) 3×3	96.0(1.2) 3×3	96.0(1.2) 3×2	96.0(1.2) 3×2
Thyroid	92.1(2.7) 3×3	92.1(2.7) 3×3	92.1(2.7) 3×3	92.1(2.7) 3×3	92.1(2.7) 3×2	92.1(2.7) 3×2
Vowel	59.0(2.8) 11×55	45.6(3.4) 11×11	29.7(2.1) 11×11	45.1(2.4) 11×11	66.7(2.6) 11×10	67.4(2.1) 12.1×11.2
Ecoli	77.8(1.7) 8×28	75.5(1.9) 8×8	79.7(1.7) 8×8	53.9(1.6) 8×8	77.0(1.5) 8×7	78.2(1.9) 9×10.1
Iris	93.3(2.2) 3×3	93.3(2.2) 3×3	93.3(2.2) 3×3	93.3(2.2) 3×3	93.3(2.2) 3×2	93.3(2.2) 3×2
Yeast	49.1(1.5) 10×45	41.3(1.2) 10×10	46.9(1.8) 10×10	39.5(1.4) 10×10	51.8(1.3) 10×9	52.0(1.2) 10.7×12.3
Glass	67.1(3.1) 7×21	60.2(3.7) 7×7	52.7(4.0) 7×7	59.2(3.7) 7×7	66.5(2.8) 7×6	66.8(2.6) 8×8.1

Table 3.16: UCI repository experiments for *NMC*.

Problem	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Balance	78.1(5.2) 3×3	77.7(4.7) 3×3	77.6(5.1) 3×3	70.3(5.7) 3×3	78.1(4.2) 3×2	79.3(4.8) 6.4×9.3
Wine	71.3(4.9) 3×3	67.9(4.7) 3×3	66.8(3.7) 3×3	69.1(3.9) 3×3	92.6(3.3) 3×2	96.7(3.2) 7.2×11.5
Thyroid	88.9(2.0) 3×3	81.8(2.1) 3×3	83.1(2.6) 3×3	80.0(2.3) 3×3	83.5(2.1) 3×2	93.7(2.2) 5.1×6.4
Vowel	58.1(2.3) 11×55	47.2(4.1) 11×11	38.6(5.3) 11×11	35.7(3.6) 11×11	54.5(3.6) 11×10	63.9(3.7) 22.3×24.7
Ecoli	82.5(2.0) 8×28	64.9(1.8) 8×8	73.4(2.2) 8×8	65.0(2.5) 8×8	83.1(2.5) 8×7	84.5(2.8) 14.3×26.8
Iris	92.6(1.6) 3×3	78.6(2.2) 3×3	78.3(2.4) 3×3	59.3(3.6) 3×3	91.1(3.0) 3×2	94.0(2.8) 6.3×9.5
Yeast	52.0(2.4) 10×45	48.8(2.3) 10×10	44.0(2.7) 10×10	48.7(2.7) 10×10	49.0(3.1) 10×9	49.2(2.8) 13.2×14.4
Glass	41.9(4.8) 7×21	25.1(4.4) 7×7	30.8(4.8) 7×7	35.6(5.2) 7×7	48.8(4.0) 7×6	66.9(3.8) 16.6×29.4

These two base classifiers obtain the least and most performance improvements, respectively. Although the results for Adaboost show that the sub-class approach is comparable with the other ECOC approaches, it can not be considered statistically significantly better. It is caused by the fact that Adaboost is a relatively strong

[10]It is important to save the history of splits to re-use the sub-groups if they are required again. It speeds up the method and also reduces the variation in the results induced by different random initializations of k-means.

Table 3.17: UCI repository experiments for *FLDA*.

Problem	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Balance	80.7(4.8) 3×3	80.7(4.8) 3×3	80.7(4.8) 3×3	80.7(4.8) 3×3	80.7(4.8) 3×2	80.7(4.8) 3×2
Wine	96.5(2.9) 3×3	96.5(2.9) 3×3	96.4(3.0) 3×3	96.5(2.9) 3×3	96.7(2.7) 3×2	96.7(2.7) 3×2
Thyroid	94.8(3.4) 3×3	88.8(4.0) 3×3	90.6(4.3) 3×3	88.8(4.5) 3×3	86.0(3.7) 3×2	93.8(3.6) 5.3×6.4
Vowel	70.3(3.8) 11×55	45.1(4.7) 11×11	46.5(4.3) 11×11	41.6(3.1) 11×11	67.5(2.8) 11×10	74.2(3.2) 19.9×32.8
Ecoli	85.2(3.5) 8×28	77.5(3.4) 8×8	78.3(3.9) 8×8	49.8(3.8) 8×8	85.2(3.4) 8×7	85.2(3.4) 8×7
Iris	98.0(2.0) 3×3	91.3(3.3) 3×3	93.3(3.4) 3×3	66.6(1.3) 3×3	97.7(2.1) 3×2	97.7(2.1) 3×2
Yeast	51.8(2.8) 10×45	30.0(3.3) 10×10	46.0(3.6) 10×10	45.5(3.4) 10×10	51.3(2.1) 10×9	51.3(2.1) 10×9
Glass	60.0(4.6) 7×21	46.8(4.9) 7×7	51.8(3.9) 7×7	53.3(4.2) 7×7	61.1(3.2) 7×6	63.0(3.7) 8.8×10.5

Table 3.18: UCI repository experiments for Linear *SVM*.

Problem	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Balance	84.6(3.2) 3×3	85.5(3.1) 3×3	85.5(3.1) 3×3	85.5(3.1) 3×3	85.5(3.1) 3×2	85.5(3.1) 3×2
Wine	93.7(1.7) 3×3	93.2(1.6) 3×3	93.2(1.6) 3×3	93.2(1.6) 3×3	95.5(1.4) 3×2	98.0(1.6) 8.7×11.3
Thyroid	94.3(2.1) 3×3	94.3(2.1) 3×3	94.3(2.1) 3×3	94.3(2.1) 3×3	94.3(2.1) 3×2	94.3(2.1) 3×2
Vowel	65.9(3.6) 11×55	32.8(2.1) 11×11	31.1(3.0) 11×11	35.7(2.5) 11×11	58.8(3.4) 11×10	68.7(2.1) 16.3×21.8
Ecoli	78.6(2.4) 8×28	68.8(3.4) 8×8	71.5(2.7) 8×8	68.3(3.8) 8×8	79.2(2.4) 8×7	81.5(2.4) 11.5×14.7
Iris	97.3(1.0) 3×3	97.3(1.0) 3×3	97.3(1.0) 3×3	97.3(1.07) 3×3	97.3(1.0) 3×2	97.3(1.0) 3×2
Yeast	51.1(4.2) 10×45	17.0(3.4) 10×10	40.5(1.2) 10×10	34.1(1.7) 10×10	51.1(2.6) 10×9	53.5(2.5) 17.8×26.1
Glass	55.5(3.2) 7×21	41.3(6.4) 7×7	37.7(2.8) 7×7	44.3(2.1) 7×7	63.8(3.1) 7×6	66.9(2.8) 10.7×13.2

Table 3.19: UCI repository experiments for *RBF SVM*.

Problem	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Balance	80.4(3.2) 3×3	69.2(3.9) 3×3	71.0(3.5) 3×3	69.2(2.9) 3×3	83.0(3.8) 3×2	83.0(3.8) 3×2
Wine	39.9(0.8) 3×3	33.1(1.0) 3×3	33.1(1.3) 3×3	33.1(1.0) 3×3	35.8(1.1) 3×2	90.8(3.1) 4.5×6.2
Thyroid	90.7(1.0) 3×3	89.8(1.6) 3×3	90.7(1.0) 3×3	90.7(1.0) 3×3	91.7(1.2) 3×2	93.7(1.3) 3.6×4.1
Vowel	82.5(2.1) 11×55	52.5(2.2) 11×11	72.2(3.6) 11×11	47.9(3.8) 11×11	73.6(3.2) 11×10	75.9(2.4) 12.4×13.5
Ecoli	80.1(3.2) 8×28	78.7(4.4) 8×8	84.2(2.8) 8×8	75.2(3.1) 8×8	82.2(3.2) 8×7	84.2(3.8) 9.8×10.3
Iris	96.0(2.8) 3×3	96.0(2.8) 3×3	96.0(2.8) 3×3	96.0(2.8) 3×3	96.0(2.8) 3×2	96.0(2.8) 3×2
Yeast	52.1(2.5) 10×45	45.5(3.2) 10×10	52.4(2.9) 10×10	46.7(2.7) 10×10	51.6(2.1) 10×9	53.2(3.2) 12.1×14.7
Glass	64.7(3.5) 7×21	51.0(3.4) 7×7	64.7(3.4) 7×7	37.4(2.2) 7×7	63.9(4.2) 7×6	66.1(3.2) 8.5×10

classifier and it is able to fit better the problem boundaries. On the other hand, looking at the results of fig. 3.16, one can see that the results of the sub-class approach

are significantly better for most of the cases because of the failure of *NMC* to model the problems by only using the original set of classes.

Statistical significance

To check for statistically significant differences between the methods, we use the Nemenyi test - two techniques are significantly different when the corresponding average ranks differ by at least the critical difference value. The ranks are obtained estimating each particular rank r_i^j for each problem i and each ECOC design j, and then, computing the mean rank R for each design as $R_j = \frac{1}{P}\sum_i r_i^j$, being P the number of experiments. The mean rank of each ECOC design for each base classifier and for the whole set of problems are numerically shown in table 3.20[11]. The critical value (CD) [22] is defined as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6P}} \tag{3.9}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$, $k = 6$ is the number of methods in the comparison, and $P = 40$ is the total number of experiments performed (8 data sets $\times$ 5 base classifiers). In our case, when comparing six methods with a confidence value $\alpha = 0.05$, $q_{0.05} = 2.20$. Substituting this in (4.28), we obtain a critical difference value of 0.81.

Observing the ranks of each ECOC design in the global rank row of table 3.20, one can observe that there are no combinations of methods for which the difference is smaller than the critical value of 0.81, and therefore, we can argue that the sub-class approach is significantly better at 95% of the confidence interval in the present experiments.

Table 3.20: Rank positions of the classification strategies for the UCI experiments.

	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
Discrete Adaboost	2.2	3.2	2.6	3.5	2.2	**1.3**
NMC	2.2	4.7	5.0	5.2	2.6	**1.1**
FLDA	1.6	3.8	3.1	3.8	2.1	**1.3**
Linear SVM	2.1	3.5	3.3	3.2	1.8	**1.0**
RBF SVM	2.3	4.2	2.6	4.3	2.6	**1.2**
Global rank	2.1	3.9	3.3	4.0	2.3	**1.2**

Parameters and training size

To show the effect of changing the parameters θ, we performed an experiment using the UCI Glass data set. In this experiment, the parameter θ_{size} is fixed to $\frac{|J|}{50}$, and the values for θ_{perf} are varied between 0.45 and 0 decreasing by 0.025 per step. For each value of θ_{perf}, the values for θ_{impr} are $\{0, 0.1, 0.2, 0.4, 0.6, 0.8, 1\}$.

The results of these experiments using NMC as the base classifier are shown graphically in fig. 3.17. In this particular problem, one can observe that until the value of $\theta_{perf} = 0.35$, the sub-class is not required since all the binary problems achieve a performance greater than this value. When this value is decreased, binary problems require the sub-class splitting approach. When θ_{impr} is increased, both

[11] We realize that averaging over data sets has a very limited meaning as it entirely depends on the selected set of problems.

the training and test performance increase. One can also observe that in the case of values near 0 for θ_{size} and near 1 for θ_{perf}, the system can specialize into very small problems, which results in overtraining. This phenomenon is just visible for $\theta_{perf} = 0.025$ and high values for θ_{impr} on the test set.

Furthermore, we analyzed the behavior of the present methodology when the training size is changed. In particular, we selected the 11-class Vowel UCI data set and performed ten-fold classification using the different ECOC designs for different training sizes: 5, 10, 20, 35, 50, 70, and 100 per cent of the training size. The base classifier in this case is also NMC. The results are shown in fig. 3.18(a). The mean number of sub-classes and binary problems is shown in the table of fig. 3.18(b). One can observe that for small training sizes the Sub-class ECOC does not split classes. At these first stages, the Sub-class ECOC becomes the DECOC strategy, and the performance is also similar, even inferior, to the one obtained by the one-versus-one strategy. When the training size is increased, though the general behavior for all the strategies is to increase their performance, the Sub-class ECOC is the one which the improvement is the most significant. Note that the performance improvement of the sub-class strategy also increases the number of sub-classes and binary problems. Still, the mean number of binary problems of 24.7 is significantly less than the 55 required for the one-versus-one strategy.

Experimental discussion

Regarding the space of parameters θ, in the present experiments the Sub-class ECOC obtains significantly better performance with fixed parameters. The fixed parameters have been chosen after a preliminary set of experiments. If it is required, one can also look for the optimum set θ which attains the best performance by using, for example, cross-validation over the training or validation sets.

Similarly, the k-means splitting criterion (that was used as a simple clustering strategy for the sake of simplicity and fast computation) can be replaced by another criterion. Suppose that we decide to keep the base classifier to be linear. In that case, it is more desirable to adapt the clustering strategy so that it guarantees the linear separability of the new clusters.

In fig. 3.19, we performed the classification of UCI data sets for the Sub-class strategy and the different base classifiers changing the splitting criterion. We compared the k-means splitting with a hierarchical clustering and the Graph Cut clustering [85]. The two clusters of the hierarchical tree are estimated using Euclidean distance to centroid for linkage. The results show that the behavior of three strategies are very similar, and there are no significant differences on the final performances.

Obviously, when we have the new split of classes, an important consideration is to decide if they contribute to improve the accuracy of the whole system. At this point, θ_{impr} looks for the improvement of the new sub-classes respect to the previous group without splitting. We commented that the value of θ_{impr} has been selected to obtain reasonably good results after a previous set of experiments. A good selection of this parameter is crucial for a good generalization of the strategy. Note that if we fix θ_{impr} to require a high performance improvement, in some cases the system could not gain from sub-classes. On the other hand, a small improvement could make the

system to focus on very small problems witch really do not contribute to the whole performance and that can produce overtraining.

To look for the minimum θ_{impr}, the implemented heuristic is based on the errors of the confusion matrix for the split groups. It provides a fast computation. Of course, the heuristic may be not optimal, and different strategies can be used instead. A natural way to deal with this problem is to look for the improvement of the whole system when including the new split problems (thus, evaluating the whole Sub-class ECOC system each time that a new problem is tested to be included). Testing this heuristic, we found that the obtained performance was very similar to the one obtained by the former approach, still maintaining a similar final number of sub-classes and binary problems, but considerably increasing the computational cost.

The sub-class technique also presents an alternative to the use of complex base classifiers. As a consequence of applying the Sub-class strategy, in the worst case it remains the same than without using sub-classes. One of the important points is that both, base classifier and sub-class, can be optimized. If the base classifier is well tuned, less binary problems and sub-classes would be required by the Sub-class strategy. On the other hand, the Sub-class approach could be seen as an incremental tool independent of the base classifier to improve the weakness of the base classifiers. For example, none of the variants of Adaboost with decision stumps is able to model the XOR problem shown in fig. 3.20(a). Fig. 3.20(b) shows the first splitting of classes found by the Sub-class strategy. In this case, Adaboost is able to model a solution considering partitions of problems using the three new sub-classes. Thus, we can take advantage from both, optimizing a base classifier and optimizing the Sub-class approach.

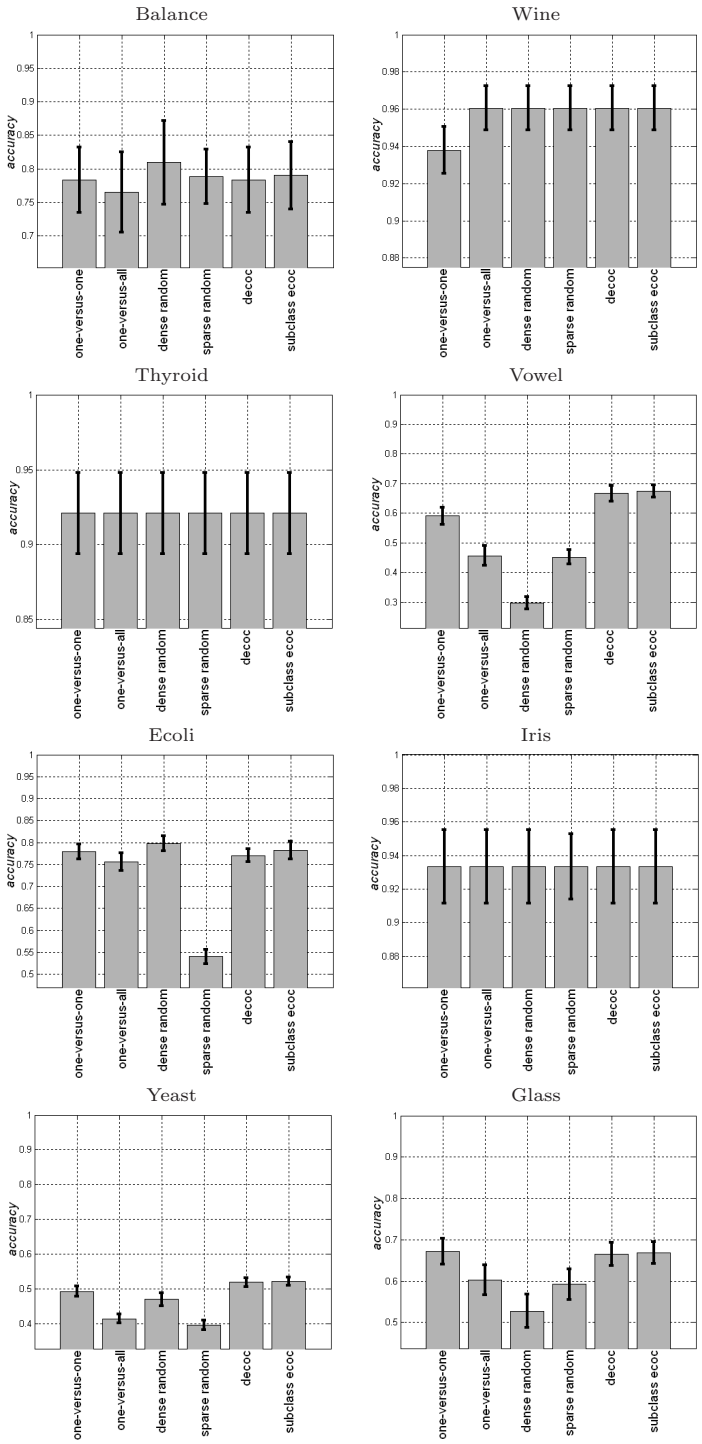


Figure 3.15: UCI experiments for Discrete Adaboost.

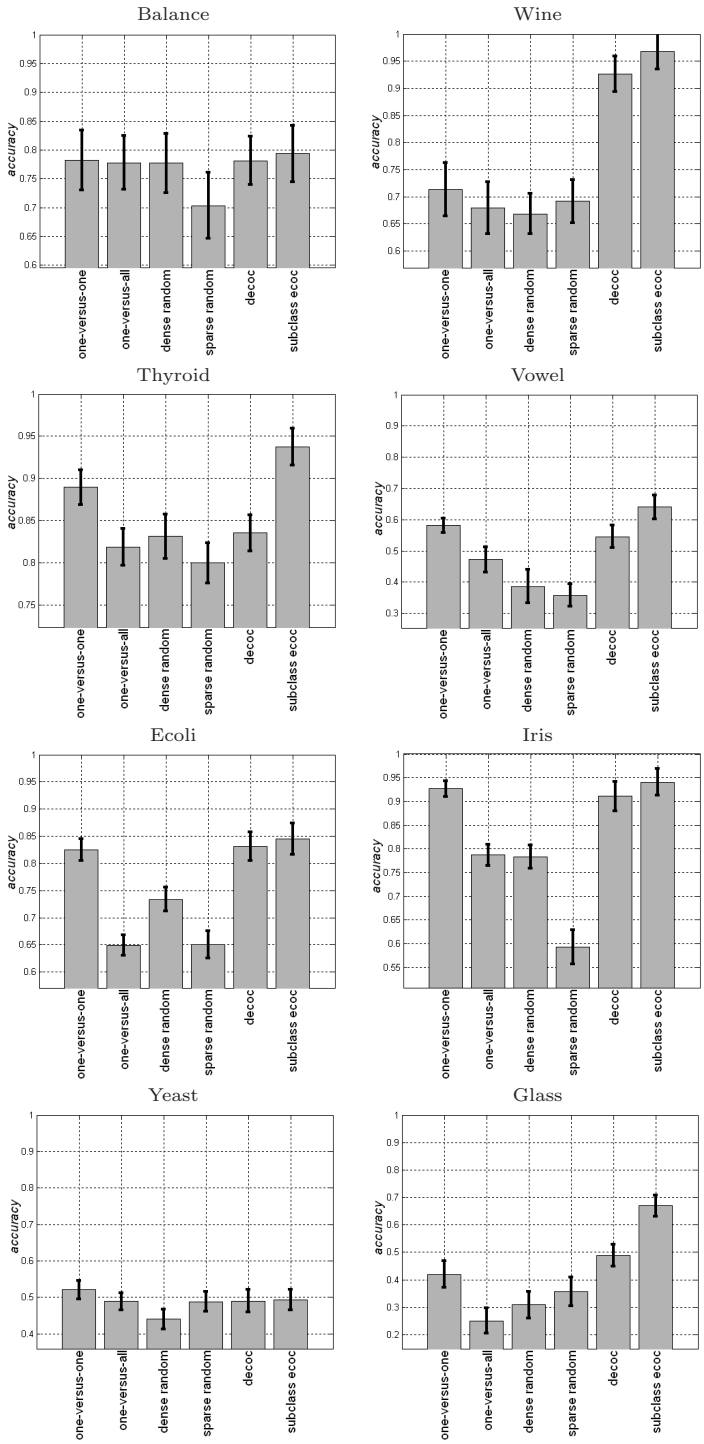


Figure 3.16: UCI experiments for *NMC*.

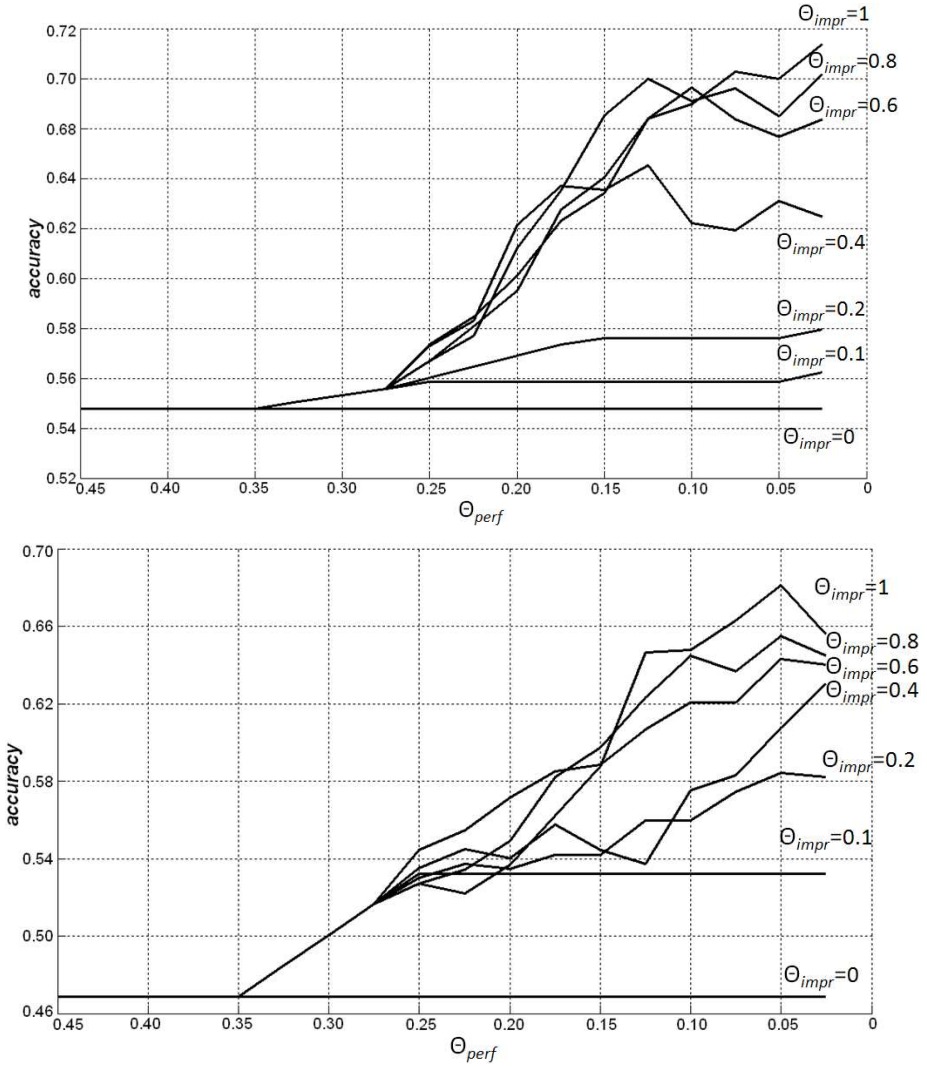
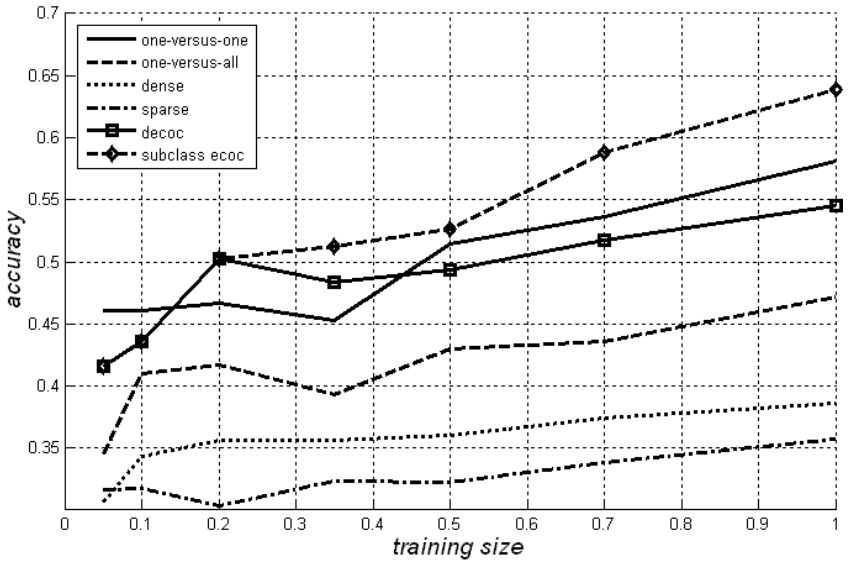


Figure 3.17: Comparison of the Sub-class ECOC performances using NMC on the UCI Glass data set for different parameters θ_{perf} and θ_{impr}. Top: training set, bottom: test set.



(a)

Training size (%)	5	10	20
Sub-classes × binary problems	11×10	11×10	11.5×10.3
35	50	70	100
12.3×13.7	15.5×16.4	18.2×19.1	22.3×24.7

(b)

Figure 3.18: (a) Test performances for the Vowel UCI data set for different percentages of the training size. (b) Mean number of sub-classes and binary problems estimated by the Sub-class ECOC for each training size. The confidence intervals of the results are between 1% and 2%.

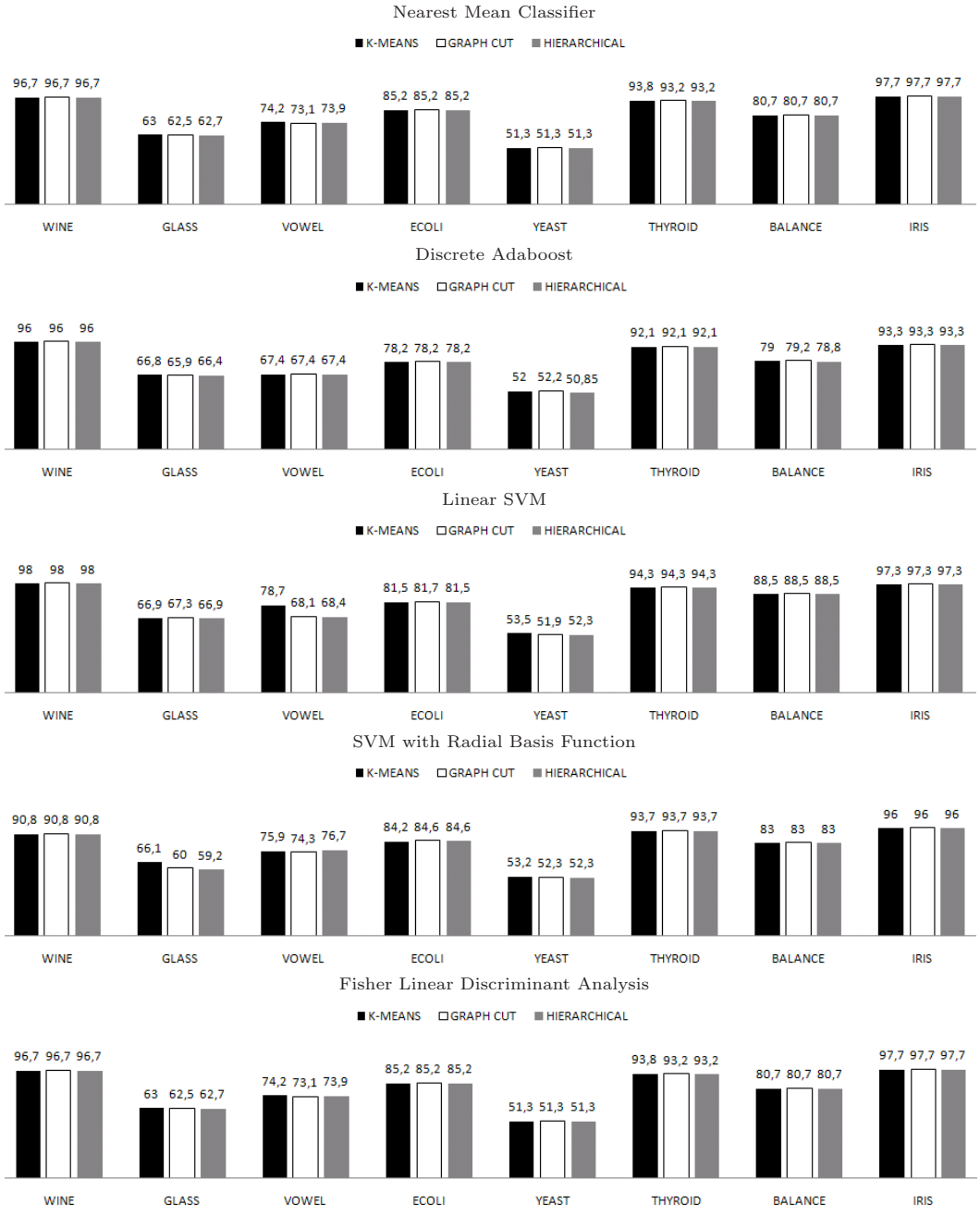


Figure 3.19: Classification performance on UCI data sets for Sub-class ECOC strategy with different splitting criteria.

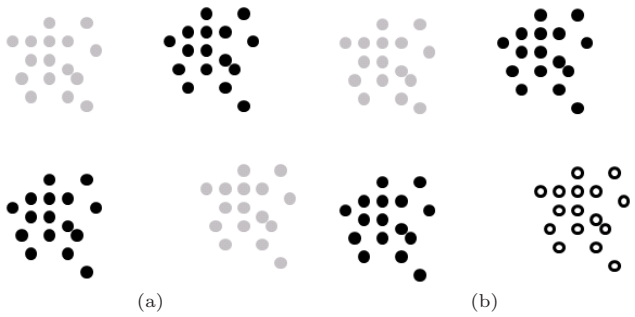


Figure 3.20: (a) Original distribution of data for two classes. (b) First Sub-class splitting.

3.4 Problem-dependent ECOC discussion

In this chapter, we presented problem-dependent designs of Error-Correcting Output Codes. The first tentative to problem-dependent design was the Forest-ECOC approach. The main idea of the method was to define binary tree structures based on the knowledge of the problem-domain. Then, the internal nodes of the tree structures are embedded as binary problems in the ECOC coding matrix. This technique can be applied to any kind of multi-class problem, yielding a small code length. The idea of trying to obtain sub-optimal tree structures as balanced as possible instead of using of the possible combinations of sub-groups of classes provides a faster way to obtain the problem-dependent code. However, when the number of classes and features increases, a greedy search remains computationally expensive, and the alternative $SFFS$ proposed to speed up the ECOC construction does not assure an optimal codeword.

The second proposed problem-dependent design is an advanced point of view from the previous design. In this case, the Optimal Node Embedding procedure provides a way to progressively increase the ECOC performance based on the evaluation of the whole system using a training/validation sub sets each time that a new dichotomizer is embedded in the coding matrix. In this sense, the length of the codeword in increased in the way that a better solution for the training data is obtained. This technique can also be applied to any kind of multi-class problem, yielding a small code length. In particular, the code obtained by this coding design tends to be smaller than the one provided by the forest-ECOC strategy since it only considers those nodes of the tree that increase the performance of the system. On the other hand, the evaluation of the system at each iteration and the search for an optimal binary problem is more expensive than the forest-ECOC approach, even when using similar sub-optimal alternatives to look for the next dichotomizer of the coding matrix.

Finally, the Sub-class ECOC approach avoids the limitations produced by the rest of ECOC designs when some distributions of the data are difficult to be modelled using some types of base classifiers due to the overlapping of the data. The Sub-class ECOC strategy splits the original set of classes into sub-classes until the base classifier is able to learn the training data or the stopping criteria are accomplished. This strategy is useful when one can not guarantee that the ECOC base classifier is able to model the binary problems of the coding matrix. In this sense, it avoids the requirement of using complex classifiers and spending several time tuning parameters. On the other hand, in the case where sub-classes are not required, this strategy remains as the DECOC approach, which could offer inferior performances to those obtained with the two previous coding alternatives presented in this chapter.

Chapter 4

ECOC Decoding

Once a problem-dependent coding matrix is learnt applying a base classifier, a decoding strategy should be applied in order to obtain a classification decision. Literature is full of binary decoding strategies, however, when we need to deal with a 3-symbol decoding, the rules of the traditional binary strategies can not be applied to the ternary case. In this section, we first overview the ternary ECOC framework. We show examples where the use of the traditional decoding strategies are inconsistent to deal with a successful classification. Second, we give a general representation of decoding strategies, from which some general properties are obtained. The properties are analyzed for the state-of-the-art decoding strategies on the new representation, and the techniques are grouped based on the properties they fulfill. Finally, different decoding strategies are proposed to deal with a successful decoding.

4.1 Ternary decoding analysis

In order to work with the large set of binary problems of the ternary ECOC framework, we need to know how to decode a ternary ECOC matrix, $M \in \{-1, 0, +1\}$. Although standard decoding strategies are currently applied over 3-symbol matrices, it seems reasonable to analyze if the traditional decoding rules are correctly used in the ternary case. To show the behavior of the standard Hamming decoding strategy of the ternary ECOC framework, we designed the example of fig. 4.1(a). In this example, a ternary coding matrix for a 7-class problem $\{c_1, .., c_7\}$ is codified by means of seven dichotomizers $\{h_1, .., h_7\}$. Observe that the first dichotomizer h_1 splits class c_1 from the rest of classes. To distinguish among classes $\{c_2, .., c_7\}$, the dichotomizers $\{h_2, .., h_7\}$ codify an one-versus-all strategy.

Now, let us observe the test codeword x of fig. 4.1(a) obtained by applying the seven dichotomizers $\{h_1, .., h_7\}$ of the coding matrix M to a new data sample ρ. The values of the codeword correspond to $x = \{1, -1, -1, -1, -1, -1, -1\}$. As commented, the test codeword can not contain the zero symbol since each classifier should vote in a way. In the example, the Hamming decoding takes as input the test codeword x and each class codeword y_i, $i \in \{1, .., 7\}$. The decoding measure obtained for each class is shown on the right of the matrix. Codeword y_1 matches the first position of x and its value is due to the six positions set to zero. When compared to the other codewords, the measure value is due to two failures between each base codeword and the test one. The output of the HD strategy assigns a higher decoding value to class c_1 in comparison to the other classes, and thus, classes $\{c_2, .., c_7\}$ are selected as the first choice.

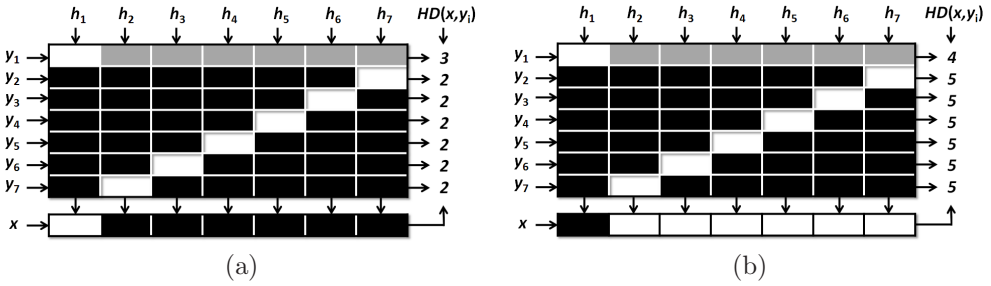


Figure 4.1: Ternary coding matrices for a 7-class problem codified using seven dichotomizers $\{h_1, .., h_7\}$. A new test codeword x is classified using the Hamming decoding.

To analyze this example, let us have a look at the sub-set of codewords represented in the *coding space* of fig. 4.2. A zero symbol in a class code introduces *one degree of freedom*, that means that both +1 and -1 are possible values during the test classification since the class has not been taken into account to train the corresponding dichotomizer. Any codeword y_i containing the zero symbol defines an extended set of possible codewords that could be obtained by examples of the class c_i. In this sense, the codeword $y_1 = \{1, 0, 0\}$ represented by the *plane* $\pi = y_1^1 = 1$ in the figure can be disambiguated into its extended set of codewords

$Y_1^e = \{\{1,1,1\},\{1,1,-1\},\{1,-1,1\},\{1,-1,-1\}\}$, where each of the four codewords of y_1 is a possible representation[1] of the same codeword y_1. Now observe the codeword $y_2 = \{1,1,1\}$ shown in the figure. Note that y_2 corresponds to one of the four representations of y_1 ($y_2 \in Y_1^e$). In the figure, y_2 then corresponds to a point in the previous *plane* π. Taking into account this decomposition, the test codeword x of fig. 4.1(a) is a possible representation of codeword y_1 of class c_1. Thus, it seems reasonable to classify x as c_1. However, in the example c_1 is the last choice. One can see this effect occurs because the decoding value increases with the number of positions that contain the zero symbol when we use the HD strategy. Let us introduce a term to denote this phenomenon:

Definition 1: ***Decoding bias*** *is the value introduced by the comparison of two codewords on positions containing the zero symbol (being the magnitude of the value proportional to the number of zero positions).*

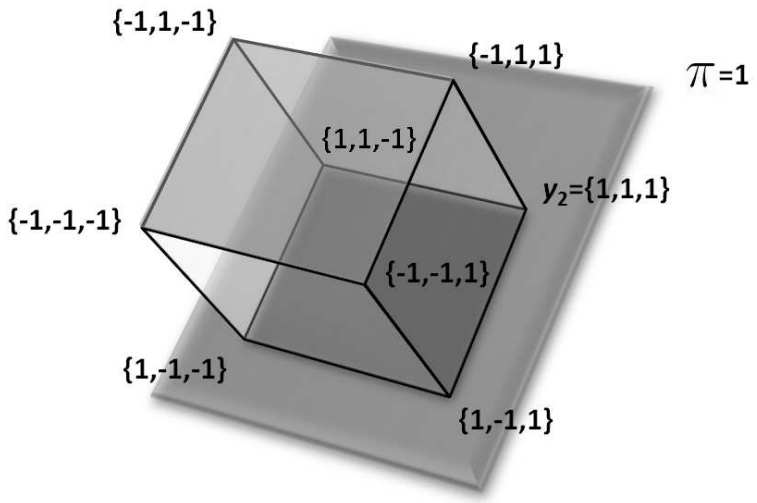


Figure 4.2: Cube of codewords of length $n = 3$.

Now observe the example of fig. 4.1(b). A new test codeword $x = \{-1,1,1,1,1,1,1\}$ is evaluated in the same ECOC design. In this case, the classification decision obtained by the HD is class c_1 with a minimum decoding value of four, while the decoding value of the rest of classes is five. Observe that the only trained classifier that takes into account c_1 is h_1. However, if we use the HD, we are deciding class c_1 according to the information obtained from the classifiers that have not considered class c_1 in their learning process. Therefore, all the information provided by class c_1 is contained in the first position of its codeword y_1.

In the example of fig. 4.1(b), either considering or not the zero positions to decode, when we use the HD, the decision in both cases is class c_1. This effect can be ex-

[1]Possible representation means that any test example of class c_1 would give a codeword from Y_1^e.

plained by the fact that the amount of bits codified by $\{-1, +1\}$ introduces a second bias that makes the measures between codewords non-comparable. It is produced because the decoding process for each codeword works in a different range of values. This effect leads to another definition:

Definition 2: ***A dynamic range bias*** *corresponds to the difference among the ranges of values associated to the decoding process of each codeword.*

Observe that this range depends on the number of positions codified by zero. In fig. 4.1(b), the codeword y_1 works on a different *dynamic range* than the rest of codewords $\{y_2, .., y_7\}$. In the example of fig. 4.1, the decoding process of codeword y_1 takes a minimum value of three when the first bit matches, and a maximum value of four when the first bit fails. It means that the *dynamic range* for the codeword y_1 is [3,4]. On the other hand, codewords $\{y_2, .., y_7\}$ can take a minimum value of zero at the decoding process when all bits match, and a maximum value of seven when all bits fails, obtaining a *dynamic range* of [0,7]. When we consider the first position of y_1 to decode, a failure on that position should have the same influence as the failure at all positions containing $\{-1, +1\}$ symbols on the rest of codewords (independently of the number of zero positions). In the same way, a match on that position also must represent the same information than to match all the positions containing $\{-1, +1\}$ symbols on the rest of codewords. Then, the codewords take values from the same *dynamic range*, and the results are comparable.

4.2 Decoding decomposition

Based on the three possible symbols of the ternary ECOC framework, we define the following terms: let b be the value produced when a bit with a $\{-1, +1\}$ value is compared to a zero symbol, a the value produced by a match in a position of a codeword containing a $\{-1, +1\}$ value, and e the value introduced by an error in a position of a codeword containing a $\{-1, +1\}$ value. Then, we introduce the following definition:

Definition 3: ***A general decoding decomposition*** *to represent decoding strategies is defined as follows:*

$$d = \sum_{k \in I_b} b_k + \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j \tag{4.1}$$

where I_b, I_a, and I_e are the sets of indexes of a codeword corresponding to the zero positions, matches on $\{-1, +1\}$ values, and failures on $\{-1, +1\}$ values, respectively. Let $|I_b| = z$, $|I_a| = \alpha$, and $|I_e| = \beta$ be the number of zeros, number of matches between two codewords, and number of failures between two codewords, respectively. In this sense, the length of a codeword is $n = z + \alpha + \beta$. Note that eq.(4.1) depends on the value induced by the zero symbols (b), the failures on the positions containing $\{-1, +1\}$ values (e), and the matching on the positions containing $\{-1, +1\}$ values (a). The value b corresponds to the *bias* induced by a zero position applying a particular decoding strategy.

As a zero symbol means that the corresponding classifier is not trained over a class, considering the *decision* of this classifier to estimate the similarity of the new test example to that class does not make sense. Thus, we define the first hypothesis as follows:

Hypothesis I: The *bias* induced by a zero position applying a particular decoding strategy should be zero ($b = 0$).

Moreover, we argue that to obtain comparable results between classes codewords, each codeword of the coding matrix M should take values in the same *dynamic range*. The *dynamic range* (DR) associated to each codeword is determined as follows:

$$DR = [min(K_1, K_2), max(K_1, K_2)], K_1 = \sum_{i \in I_a} |a_i|, K_2 = \sum_{j \in I_e} |e_j| \tag{4.2}$$

If K_1 and K_2 are constant shared factors for all the codewords, the *dynamic range* is maintained for all classes, and the decoding measures are comparable. Then, we define the second hypothesis as follows:

Hypothesis II: K_1 and K_2 should be constant shared factors for all the codewords.

Based on the previous hypotheses, we define four types of decoding strategies:

Table 4.1: Types of decoding strategies.

	$b \neq 0$	$b = 0$
Different *dynamic ranges*	Type 0	Type I
Same *dynamic ranges*	Type II	Type III

Definition 4: *A decoding strategy is of* ***Type 0*** *if the bias produced by the 0 symbol is higher than zero ($b > 0$), and the dynamic ranges between codewords differ.*

Definition 5: *A decoding strategy is of* ***Type I*** *if the bias produced produced by the 0 symbol is null ($b = 0$), and the* dynamic ranges *between codewords differ.*

Definition 6: *A decoding strategy is of* ***Type II*** *if the bias produced by the 0 symbol is higher than zero ($b > 0$), and the dynamic ranges between codewords are the same.*

Definition 7: *A decoding strategy is of* ***Type III*** *if the bias produced by the 0 symbol is null ($b = 0$), and the dynamic ranges between codewords are the same.*

Table 4.1 summarizes the groups of decoding strategies based on the previous definitions.

4.2.1 Analysis of state-of-the-art decoding strategies

Following the introduced notation, we split the state-of-the-art decoding strategies according to the decomposition of eq.(4.1) and analyze the two previous properties in each case.

The analysis is performed over the decoding strategies reviewed in the previous section: Hamming decoding, Inverse Hamming decoding, Euclidean decoding, Attenuated Euclidean decoding, Loss-based decoding, and the Probabilistic decoding approach of [67]. To split each decoding strategy in the decomposition representation, we change the decision rules of the probabilistic strategies in order to consider eq.(4.1) as a function of the *measure* to be minimized.

• *Hamming decoding*:

We can easily find a correspondence between the original formulation of the HD and the representation of eq.(4.1). HD always includes a value of $\frac{1}{2}$ for b_k, $k \in I_b$. A match does not take influence in the measure ($a_i = 0$, $i \in I_a$), and a failure on a position increases the measure in $e_j = 1$, $j \in I_e$. Then, the new representation can be defined as follows:

$$HD(x,y) = \sum_{k \in I_b} b_k + \sum_{j \in I_e} e_j = \frac{z}{2} + \beta \tag{4.3}$$

Analyzing the Hamming decoding in the ternary case, one can observe that the zero positions introduce a *bias* of $\frac{z}{2}$. Moreover, the prediction is influenced by the value of z, which makes codewords take values from different *dynamic ranges* for different number of zero positions. In this sense, HD corresponds to the strategies of Type 0.

• *Inverse Hamming decoding:*

Looking at eq. (2.3), the term Δ^{-1} of the IHD corresponds to a constant factor dependent of the class codes. Therefore, we can fix on the term D^T to find a corre-

spondence to eq. (4.1). The term $\Delta_1^{-1}D^T$ stands for the IHD for the first codeword of the coding matrix M. Note that Δ_1^{-1} does not depend on the test codeword x. Then, if the components of the first row of Δ^{-1} correspond to $\{W_1, .., W_N\}$, the result of the product $\Delta_1^{-1}D^T$ can be defined as follows:

$$IHD(x, y_1) = \Delta_1^{-1}D^T = \sum_{j=1}^{N} W_j \cdot HD(x, y1) = \sum_{j=1}^{N} W_j \left(\frac{z_j}{2} + \beta_j\right) \tag{4.4}$$

which implies:

$$IHD(x, y_1) = z_1 \frac{1}{2}\left(W_1 + \sum_{i=2}^{N} \frac{W_i z_i}{z_1}\right) + \beta_1 \left(W_1 + \sum_{i=2}^{N} \frac{W_i \beta_i}{\beta_1}\right) \tag{4.5}$$

This expression exactly corresponds to the representation of eq. (4.1) for a codeword y_1, where $b_k = -1\left(W_1 + \sum_{i=2}^{N} \frac{W_i z_i}{z_1}\right), k \in I_b$, $e_j = -1\left(W_1 + \sum_{i=2}^{N} \frac{W_i \beta_i}{\beta_1}\right), j \in I_e$, and $a_i = 0, i \in I_a$, being the weights W dependent on the design of the coding matrix M. Note that different *bias* are induced by different number of zeros, and different *dynamic ranges* are also obtained for different values of z and β. Thus, the IHD corresponds to the Type 0 strategies.

• *Euclidean decoding:*

The parameters in this case are: $b_k = 1, k \in I_b$, $a_i = 0, i \in I_a$, and $e_{=}4, j \in I_e$, obtaining the following representation:

$$ED(x, y) = \sum_{k \in I_b} b_k + \sum_{j \in I_e} e_j = \sum_{k \in I_b} 1 + \sum_{j \in I_e} 4 = z + 4\beta \tag{4.6}$$

Compared to the error induced by the zero symbol in the HD strategy, one can observe that in this case, $b_k, k \in I_b$ is less significative in comparison with $e_j, j \in I_e$. In fig. 4.3 one can see this behavior for different number of zero positions. When the number of zeros increases, the error accumulated by the ED is less significative than the HD error. This is one of the main reasons why the ED usually improves the performance of the HD asymptotically when applied to ternary symbol-based ECOC [73]. This strategy also is of Type 0.

• *Loss-based decoding:*

We introduce the Loss-Based decoding in the representation of eq.(4.1) for the Linear Loss-based function and the Exponential Loss-based function. Using a Loss-function, the final measure is obtained by means of an additive model where the matches introduce negative weights. In particular, the LLB parameters are as follows: $b_k = 0, k \in I_b$, $a_i = -1, i \in I_a$, and $e_j = 1, j \in I_e$, giving:

$$LLB(x, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} (-1) + \sum_{j \in I_e} 1 = -\alpha + \beta \tag{4.7}$$

The ELB parameters are: $b_k = 1, k \in I_b$, $a_i = 1/e, i \in I_a$, and $e_j = e, j \in I_e$, obtaining:

$$ELB(x, y) = \sum_{k \in I_b} b_k + \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{k \in I_b} 1 + \sum_{i \in I_a} 1/\mathbf{e} + \sum_{j \in I_e} \mathbf{e} = z + \frac{\alpha}{\mathbf{e}} + \beta\mathbf{e} \tag{4.8}$$

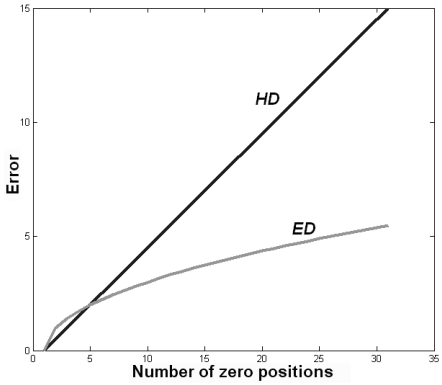


Figure 4.3: Errors induced by the zero symbol for the HD and ED decoding strategies.

In the case of LLB, the matches and the failures have the same influence, while in the ELW, a failure is $\mathbf{e}^2$ times more significant than a match. One can see that $b_k = 0, k \in I_b$ by LLB, but in both LLB and ELB, different *dynamic ranges* are obtained for different values of α and β for LLB, and z, α, and β for ELB. Thus, ELB is of Type 0 and LLB of Type I.

• Probabilistic decoding:

From the initial definition of this strategy (eq.(2.6)), we can fix the parameters $K = \omega = 0$ and $\upsilon = 1$ in order to simplify the study of the technique, which leads to the following equation:

$$PD(x,y) = -\log\left(\left(\frac{1}{1+\mathbf{e}}\right)^{\alpha}\left(\frac{1}{1+1/\mathbf{e}}\right)^{\beta}\right) \tag{4.9}$$

We can easily change this representation into the form of eq. (4.1) by defining: $b_k = 0, k \in I_b$, $a_i = \log\left(\frac{1}{1+\mathbf{e}}\right), i \in I_a$, $e_j = \log\left(\frac{1}{1+1/\mathbf{e}}\right), j \in I_e$, which implies:

$$\begin{aligned} PD(x,y) &= \sum_{i\in I_a} a_i + \sum_{j\in I_e} e_j \\ PD(x,y) &= \sum_{i\in I_a} \log\left(\frac{1}{1+\mathbf{e}}\right) + \sum_{j\in I_e} \log\left(\frac{1}{1+1/\mathbf{e}}\right) \\ PD(x,y) &= \alpha \log\left(\frac{1}{1+\mathbf{e}}\right) + \beta \log\left(\frac{1}{1+1/\mathbf{e}}\right) \end{aligned}$$

This strategy was proposed to deal with the ternary decoding. In particular, it satisfies that $b_k = 0, k \in I_b$ since the induced *bias* by the zero symbol is null. However, note that different *dynamic ranges* are obtained for different values of α and β, and thus, the strategy is of Type I.

4.3 Attenuated Euclidean Decoding

This technique is an adaptation of the Euclidean decoding that takes into account Hypothesis I. The formulation is redefined taking into account the factors $\mid y_i^j \mid\mid x^j \mid$; that makes the measure to be unaffected by the positions of the codeword y_i that contain the zero symbol ($\mid y_i^j \mid = 0$). Note that in most of the cases $|x^j| = 1$. Then, the Euclidean Decoding measure is redefined as follows:

$$AED(x, y_i) = \sqrt{\sum_{j=1}^{n} \mid y_i^j \mid\mid x^j \mid (x^j - y_i^j)^2} \tag{4.10}$$

In this case, the difference between the ED and the AED is the value of $b_k, k \in I_b$, fixed to zero by AED. The new representation is as follows:

$$ED(x, y) = \sum_{j \in I_e} e_j = \sum_{j \in I_e} 4 = 4\beta \tag{4.11}$$

Note that the weighting parameter of AED avoids the *bias* produced by the zero symbol. Nevertheless, different *dynamic ranges* are obtained for different values of β. This strategy corresponds to Type I.

4.4 Laplacian and Pessimistic β-Density Distribution Decoding

Based on the presented grouping of strategies, none of the decoding techniques in the literature belongs to Type II or Type III. In this section, we introduce two novel decoding strategies of Type III. First, we propose a methodology based on the discrete output of the classifiers, called Pessimistic β-Density Distribution decoding ($\beta - DEN$). After that, we extend its behavior using a continuous extension.

The simplest way to avoid the *bias* of the third symbol is to ignore the positions coded by zero. This yields in a measure that counts the number of coincidences between the input codeword and the class codeword. In order to make all the codewords to work in the same *dynamic range*, the measure is normalized by the total number of positions coded by $\{-1, +1\}$, obtaining $d(x, y_i) = \alpha_i/(\alpha_i + \beta_i)$. The main drawback of this definition is that it is not robust when there is a small number of coded positions in one word. In order to alleviate this problem, we introduce a prior bias, known as the Laplace correction. With this correction, the new decoding score, called Laplacian decoding (LAP), is defined as follows:

$$LAP(x, y_i) = \frac{\alpha_i + 1}{\alpha_i + \beta_i + K} \tag{4.12}$$

where K is an integer value that codifies the number of classes considered by the classifier - two in this case.

Based on this formulation, we can define a sub-optimal method, called Pessimistic β-Density Distribution decoding. The method is based on estimating the probability density functions between two codewords. The main goal of this strategy is to model at the same time the accuracy and uncertainty based on a pessimistic score in order to obtain more reliable predictions. We use an extension of the continuous binomial distribution, the β-distribution, defined as follows:

$$\psi_i(\nu, \alpha_i, \beta_i) = \frac{1}{K}\nu^{\alpha_i}(1 - \nu)^{\beta_i} \tag{4.13}$$

where ψ_i is the β-Density Distribution between a codeword x and a class codeword y_i for class c_i, and $\nu \in [0, 1]$. The expectation of ψ_i is $\alpha_i/(\alpha_i + \beta_i)$. Note that it asymptotically tends to the Laplace corrected estimator without the prior K in eq.(4.12).

Given a test codeword x and the set of functions $\psi(\nu, \alpha, \beta) = [\psi_1(\nu, \alpha_1, \beta_1)$,..., $\psi_N(\nu, \alpha_N, \beta_N)]$, the class c_i is assigned to x if it achieves the highest score s_i, defined as the pessimistic score satisfying the following equivalency:

$$s_i \; : \quad \int_{\nu_i - s_i}^{\nu_i} \psi_i(\nu, \alpha_i, \beta_i) d\nu = u \tag{4.14}$$

where u is a threshold parameter. After a preliminary set of experiments, we fixed $u = \frac{1}{3}$. Note that u governs the uncertainty influence in the final score. Figure 4.4 shows the estimated density functions $[\psi_1, \psi_2, \psi_3, \psi_4]$ for the design shown in fig. 2.1(b). Observe that on the design of fig. 2.1(b), the HD and the ED decoding

strategies classify the test codeword x by class c_1, although the decision should be class c_2. In fig. 4.4, one can see that the β-DEN decoding classifies the test data sample to its correct class c_2, obtained by fig. 4.4(b). It can be shown that when a function ψ_i is estimated by a combination of values α_i and β_i, the sharpness is higher when it is generated by a majority of one of the two types. Besides, this sharpness depends on the number of code positions different to zero and the balance between the number of matches and failures. In this way, the pessimistic score reflects the confidence in the expectation of the probability density function.

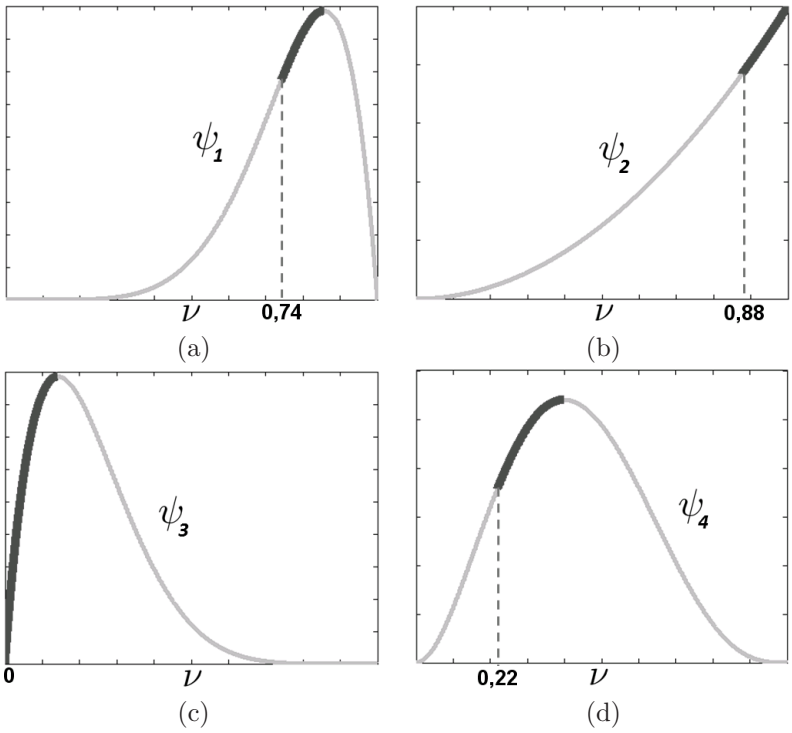


Figure 4.4: Pessimistic Score decoding for the test codeword x and the matrix M for the four classes of fig. 2.1(b). (a) Class c_1, (b) class c_2, (c) class c_3, and (d) class c_4. The probability for the second class allows a successful classification in this case.

Now, let us analyze the β-Density to obtain the representation in the form of eq.(4.1). We can apply minus logarithm to the β-Density formulation to split it. We obtain the parameters: $b_k = 0, k \in I_b$, $a_i = -\log(\nu), i \in I_a$, and $e_j = -\log(1-\nu), j \in I_e$, and the following representation:

$$\beta - DEN(x,y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} (-\log(\nu)) + \sum_{j \in I_e} (-\log(1-\nu)) = -\alpha \log(\nu) - \beta \log(1-\nu) \quad (4.15)$$

Note that in the β-DEN decoding the zero symbol has no influence, and the *dynamic range* for all the codewords takes values in the same interval $[0, 1]$, being a strategy of Type III.

4.5 Loss-Weighted Decoding

We define the novel Loss-Weighted decoding based on a combination of normalized probabilities to adapt the ternary ECOC decoding to the Type III strategies. The properties are encoded in a matrix that is used to weight the decoding process. The weight matrix codifies hypothesis I and II, being independent of the coding and decoding strategy applied. Moreover, as not all the hypotheses have the same performance on learning the data samples, the accuracy of each binary problem is used to adjust the final classification decision.

We define a weight matrix M_W, by assigning to each position of the codeword codified by $\{-1, +1\}$ a weight of $\frac{1}{n-z}$. As $\alpha + \beta = n - z$, by excluding the zero-positions, the previous process codifies the same *dynamic range* for all codewords. Moreover, the *bias* of the third symbol is avoided by assigning a weight of zero to those positions of the weight matrix M_W that contain a zero in the coding matrix M. In this way, $\sum_{j=1}^{n} M_W(i,j) = 1, \forall i = 1, ..., N$, satisfying Type III properties.

We assign to each position (i,j) of a performance matrix H a continuous value that corresponds to the performance of the dichotomizer h_j classifying the samples of class c_i as follows:

$$H(i,j) = \frac{1}{m_i} \sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i), i, j), \quad \text{based on} \quad \varphi(x^j, i, j) = \begin{cases} 1, & \text{if} \quad x^j = y_i^j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.16}$$

Note that eq.(4.16) makes H to have zero probability at those positions corresponding to unconsidered classes.

We normalize each row of the matrix H so that M_W can be considered as a discrete probability density function:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)}, \quad \forall i \in [1, ..., N], \quad \forall j \in [1, ..., n] \tag{4.17}$$

In fig. 4.5, a weight matrix M_W for a 3-multi-class problem of four hypotheses is estimated. Figure 4.5(a) shows the coding matrix M. The matrix H of fig. 4.5(b) represents the accuracy of the hypotheses classifying the instances of the training set. The normalization of H results in a weight matrix M_W shown in fig. 4.5(c).

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix} \quad H = \begin{bmatrix} 0.955 & 0.955 & 1.000 & 0.000 \\ 0.900 & 0.800 & 0.000 & 0.000 \\ 1.000 & 0.905 & 0.805 & 0.805 \end{bmatrix} \quad M_W = \begin{bmatrix} 0.328 & 0.328 & 0.344 & 0.000 \\ 0.529 & 0.471 & 0.000 & 0.000 \\ 0.285 & 0.257 & 0.229 & 0.229 \end{bmatrix}$$

(a) (b) (c)

Figure 4.5: (a) Coding matrix M of four hypotheses for a 3-class problem. (b) Performance matrix H. (c) Weight matrix M_W.

Once we obtain the weight matrix M_W is done, we introduce the weight matrix in the Loss-based decoding. The decoding estimation is obtained by means of an ELB decoding model $L(\theta) = \mathbf{e}^{-\theta}$, where θ corresponds to $y_i^j \cdot f(\rho, j)$ (similar to the

Loss-based decoding), weighted using M_W[2]:

$$LW(\rho, i) = \sum_{j=1}^{n} M_W(i,j) L(y_i^j \cdot f(\rho, j)) \tag{4.18}$$

The summarized algorithm is shown in table 4.2.

Note that the weight matrix M_W encoding the ternary decoding properties is independent of the coding and decoding strategies applied. In this sense, It can be potentially applied to any existing decoding strategy. For the present formulation, we choose the Loss-based decoding as the base decoding strategy to apply the weight matrix M_W since LB was one of the firsts attempts to the ternary decoding. In this sense, different weighted decodings can be formulated[3]. Moreover, note that depending on the problem we are working on, not only the continuous output of the base classifier could be useful to weight the matrix M_W, but also prior information about the classes distribution (or class frequencies instead) as well as other useful information can also be included.

Table 4.2: Loss-Weighted algorithm.

Loss-Weighted strategy: Given a coding matrix M,

1) Calculate the performance matrix H:

$$H(i,j) = \frac{1}{m_i} \sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i), i, j) \quad \text{based on} \quad \varphi(x^j, i, j) = \begin{cases} 1, & \text{if } x^j = y_i^j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.19}$$

2) Normalize H: $\sum_{j=1}^{n} M_W(i,j) = 1, \quad \forall i = 1, ..., N$:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)}, \quad \forall i \in [1, ..., N], \quad \forall j \in [1, ..., n] \tag{4.20}$$

3) Given a test data sample ρ, decode based on:

$$LW(\rho, i) = \sum_{j=1}^{n} M_W(i,j) L(y_i^j \cdot f(\rho, j)) \tag{4.21}$$

To obtain the formulation of LW in the representation of eq.(4.1), we consider the use of the linear and the exponential Loss-functions with discrete and continuous possible outputs of the classifiers.

In the case of the Linear Loss-Weighted using the continuous output of the classifier LLW_C, we obtain the values $b_k = 0, k \in I_b$, $a_i = -M_W(\_, i)|f^i(\rho)|, i \in I_a$, $M_W(\_, i) \in [0, 1]$, where '_' stands for the row which corresponding codeword is being compared,

[2]Note that different Loss-functions as well as discrete and continuous outputs of the classifiers can also be applied.

[3]We have performed some experiments applying the weight matrix over other decoding strategies, such as the Weight-Euclidean decoding, obtaining significant performance improvements

and $e_j = M_W(\_, j)|f^j(\rho)|, j \in I_e$, $M_W(\_, j) \in [0, 1]$. Then, the new representation is as follows:

$$LLW_C(\rho, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} -M_W(\_, i)|f^i(\rho)| + \sum_{j \in I_e} M_W(\_, j)|f^j(\rho)| \quad (4.22)$$

And the following parameters considering a discrete output of the classifier LLW_D: $b_k = 0, k \in I_b$, $a_i = -M_W(\_, i), i \in I_a$, $M_W(\_, i) \in [0, 1]$, and $e_j = M_W(\_, j), j \in I_e$, $M_W(\_, j) \in [0, 1]$, giving:

$$LLW_D(x, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} -M_W(\_, i) + \sum_{j \in I_e} M_W(\_, j) \quad (4.23)$$

If we take as baseline the previous discrete representation of eq.(4.23), and consider each dichotomizer to properly learn the complete training data, we obtain: $b_k = 0, k \in I_b$, $a_i = -\frac{1}{n-z}, i \in I_a$, and $e_j = \frac{1}{n-z}, j \in I_e$, which implies:

$$LLW_D(x, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} \left(-\frac{1}{n-z}\right) + \sum_{j \in I_e} \frac{1}{n-z} = -\frac{\alpha}{n-z} + \frac{\beta}{n-z} \quad (4.24)$$

Then, we can observe that in the discrete LLW_D, the zero symbol is not considered. Moreover, independently of the number of positions coded by $\{-1, +1\}$, if all these positions match, then $\alpha = n - z$, and the parameter $-\frac{\alpha}{n-z}$ of eq. (4.24) is maintained constant to $K_1 = -1$ for all the codewords. In the case that all positions coded by $\{-1, +1\}$ correspond to failures, $\beta = n - z$, obtaining a constant value $K_2 = 1$. Therefore, all the codewords take values in the interval $[-1, 1]$. The same occurs with the continuous LLW_C, since the previous behavior is only affected by the factor introduced by the margin of the output of the classifier.

Applying the same formalism in the case of the continuous Exponential Loss-Weighted ELW_C, we obtain the following parameters: $b_k^{(0)} = 0, k \in I_b$, $a_i = \frac{M_W(\_, i)}{\mathbf{e}^{|f^i(\rho)|}}, i \in I_a$, $M_W(\_, i) \in [0, 1]$, and $e_j = M_W(\_, j)\mathbf{e}^{|f^j(\rho)|}, j \in I_e$, $M_W(\_, j) \in [0, 1]$, obtaining:

$$ELW_C(\rho, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} \frac{M_W(\_, i)}{\mathbf{e}^{|f^i(\rho)|}} + \sum_{j \in I_e} M_W(\_, j)\mathbf{e}^{|f^j(\rho)|} \quad (4.25)$$

And the following parameters considering a discrete output of the classifier ELW_D: $b_k^{(0)} = 0, k \in I_b$, $a_i = \frac{M_W(\_, i)}{\mathbf{e}}, i \in I_a$, $M_W(\_, i) \in [0, 1]$, and $e_j = M_W(\_, j)\mathbf{e}, j \in I_e$, $M_W(\_, j) \in [0, 1]$, obtaining:

$$ELW_D(x, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} \frac{M_W(\_, i)}{\mathbf{e}} + \sum_{j \in I_e} M_W(\_, j)\mathbf{e} \quad (4.26)$$

If we take as baseline the previous discrete representation of eq.(4.26), and consider each dichotomizer to properly learn the complete training data, we obtain: $b_k = 0, k \in I_b$, $a_i = \frac{1}{(n-z)\mathbf{e}}, i \in I_a$, and $e_j = \frac{\mathbf{e}}{n-z}, j \in I_e$, which implies:

$$ELW_D(x, y) = \sum_{i \in I_a} a_i + \sum_{j \in I_e} e_j = \sum_{i \in I_a} \frac{1}{(n-z)\mathbf{e}} + \sum_{j \in I_e} \frac{\mathbf{e}}{n-z} = \frac{\alpha}{(n-z)\mathbf{e}} + \frac{\beta\mathbf{e}}{n-z} \quad (4.27)$$

In the previous *ELW* cases, the zero symbol is not considered. If all positions coded by $\{-1,+1\}$ correspond to matches, $\alpha = n - z$, which makes all the codewords to obtain the constant value $K_1 = \frac{-1}{\mathbf{e}}$. On the other hand, if all positions coded by $\{-1,+1\}$ correspond to failures, then $\beta = n - z$ implies a constant factor $K_2 = \mathbf{e}$, which makes all codewords to obtain values in the same *dynamic range* $[-\frac{1}{\mathbf{e}}, \mathbf{e}]$. Thus, all the *LW* variants correspond to Type III strategies.

4.6 Taxonomy of decoding strategies

Table 4.3 summarizes the values of the parameters obtained using the representation of eq.(4.1) for all the decoding strategies. The decoding strategies of table 4.3 are sorted from Type 0 to Type III designs. At the first column of the table, the values of b different to zero point out the methods that introduce a *bias* for the zero symbol. Note that four of the traditional approaches do not avoid this *bias*. The columns of values a and e stands for the values introduced by a match and a fail at the decoding step, respectively. Note that none of the traditional decoding strategies presented in the literature belongs to Type II and Type III strategies since the *dynamic ranges* differ for different number of positions coded by zero. Only the $\beta - DEN$ and *LW* decoding variants normalize the *dynamic ranges* to work in the same range of values for all codewords. Note that if we substitute in eq.(4.1) the parameters b, a, and e of each decoding strategy, we obtain an equivalent decoding evaluation than using its original formulation.

Table 4.3: Decoding parameters in the decomposition of eq.(4.1).

Strategy	b	a	e
HD	$1/2$	0	1
IHD	$\frac{-1}{2}\left(W_1 + \sum_{i=2}^{N} \frac{W_i z_i}{z_1}\right)$	0	$-1\left(W_1 + \sum_{i=2}^{N} \frac{W_i \beta_i}{\beta_1}\right)$
ED	1	0	4
AED	0	0	4
LLB	0	-1	1
PD	0	$\log\left(\frac{1}{1+\mathbf{e}}\right)$	$\log\left(\frac{1}{1+1/\mathbf{e}}\right)$
ELB	1	$1/\mathbf{e}$	$\mathbf{e}$
β-*DEN*	0	$\log(\nu)$	$\log(1-\nu)$
LLW_C	0	$-M_W(\_, i)\lvert f(\rho)\rvert$	$M_W(\_, j)\lvert f(\rho)\rvert$
LLW_D	0	$-M_W(\_, i)$	$M_W(\_, j)$
ELW_C	0	$\frac{M_W(\_,i)}{\mathbf{e}^{\lvert f(\rho)\rvert}}$	$M_W(\_, j)\mathbf{e}^{\lvert f(\rho)\rvert}$
ELW_D	0	$\frac{M_W(\_,i)}{\mathbf{e}}$	$M_W(\_, j)\mathbf{e}$

Based on the previous types of decoding strategies and with the use of discrete or continuous outputs of the classifiers, six different groups of decodings are shown in table 4.4. The Laplacian decoding *LAP* has also been included as the simplest choice of Type III strategy. Some strategies, such as *ED*, *AED*, *LB*, and *PD* can also be used in both discrete and continuous domains, though there are no evidences

of their use in the literature. Note that none of the decoding strategies presented in the literature belongs to Type II strategies since it does not exists a method that maintain the *dynamic range* for all codewords at same time that includes *bias* for the zero symbol.

Table 4.4: Decoding strategies grouped by type and discrete/continuous domains.

Type	Discrete	Continuous
Type 0	*HD*, *IHD*, *ED*	-
Type I	*AED*	*LB*, *PD*
Type III	$\beta - DEN$, *LAP*, *LW*	*LW*

On the next section, we perform several experiments to test the proposed methodology. Based on the present formulation, our working hypothesis is that when the decoding strategies avoid the *bias* produced by the zero symbol and all the codewords work in the same *dynamic range*, the performance of the ECOC designs is improved. Therefore, we apply the decoding strategies on the state-of-the-art coding designs and we test their behavior over different multi-class data sets.

4.7 Decoding evaluation

Before the results are presented, we discuss our validation methodology regarding the data, comparatives, measurements, and experiments[4].

• *Data*: The data used for the experiments consists of 16 multi-class data sets from the UCI Machine Learning Repository data set [8].

• *Comparatives*: For the comparatives, we used the decoding strategies analyzed in this chapter: Hamming decoding, Euclidean decoding, Inverse Hamming decoding, Attenuated Euclidean decoding, Loss-based decoding with Linear and Exponential Loss-functions, Probabilistic decoding, Laplacian decoding as the simplest choice of Type III strategy, Pessimistic β-Density Distribution decoding, and four variants of the Loss-Weighted decoding strategy: the Linear Loss-Weighted with discrete and continuous outputs of the classifier, and the Exponential Loss-Weighted with discrete and continuous outputs of the classifier.

Furthermore, all the decoding strategies are applied over the state-of-the-art ECOC coding designs: one-versus-one [88], one-versus-all [68], dense random [5], sparse random [5], DECOC [73], and ECOC-ONE designs. The parameters of the coding strategies are the predefined or the default values given by the authors. The dense and sparse matrices are selected from a set of 20000 generated random matrices, and fixed to length of N for a further comparison with the one-versus-all, DECOC, and ECOC-ONE strategies for a similar number of dichotomizers.

• *Measurements*: To measure the performance of the different strategies we apply stratified ten-fold cross-validation and test for confidence interval at 95% with a two-tailed t-test. We also use the Nemenyi test to look for significant statistical differences between the methods performances at 95% [22]. The base classifiers used for the experiments are Gentle Adaboost with 40 runs of decision stumps [32] and the Linear OSU implementation of Support Vector Machines (SVM) [37][66].

• *Experiments*: we evaluate the classification of 16 UCI data sets.

UCI classification

The first experiment consists of classifying 16 multi-class UCI repository data sets. The details of the data sets can be found in chapter F.

In this experiment, the 13 decoding strategies are applied over the six coding designs and tested on the 16 UCI data sets for the two base classifiers, which corresponds to a total of 2496 ten-fold experiments. In order to summarize these results, we estimated the ranking of each particular decoding strategy for the two different base classifiers. Thus, each decoding strategy has been applied over six codings $\times$ 16 data sets. Using these 96 experiments for each decoding, the ranking considering the intersection of confidence intervals on one hand and without considering the confidences on the other hand are shown in fig. 4.6 for Gentle Adaboost and Linear SVM, respectively. All the performances from which the ranking are computed are also shown in the tables of Appendix D. The rankings are obtained estimating each particular ranking r_i^j for each problem i and each decoding j, and computing the mean ranking R for each decoding as $R_j = \frac{1}{J}\sum_i r_i^j$, where J is the total number of problems (6 codings $\times$ 16 data sets). Note that either for the Gentle Adaboost base classifiers and

[4]More experimental results and analysis of the decoding methodology are shown in chapter 7.

the Linear SVM classifier the ranking positions of each decoding strategy is the same in most cases. When the confidence interval is considered, the ranking differences are less significative, but the relative positions are also maintained. The general behavior of this graphics shows that Type III strategies, and in particular the four variants of the Loss-Weighted decoding, attain the best performance in the experiments, followed by Type I strategies, and finally by Type 0 strategies. Note that in the variants of LW, for both Gentle Adaboost and Linear SVM, the differences between LLW with discrete and continuous outputs of the classifier are not significant, but in the case of ELW, the performance is improved by considering the continuous values of the output of both base classifiers.

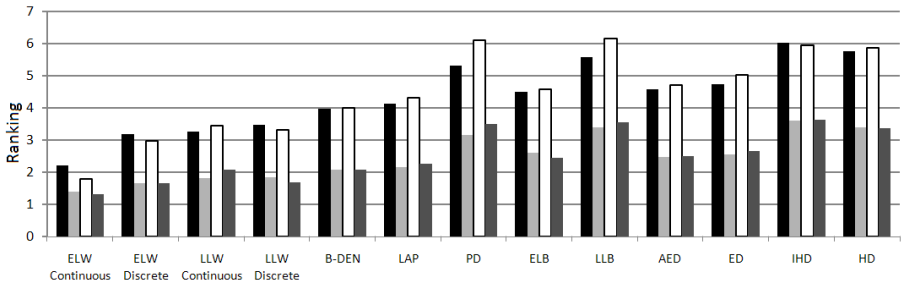


Figure 4.6: Ranking for the decoding strategies over all coding designs and UCI data sets: Gentle Adaboost without (in black) and considering (in white) the intersection of the confidence intervals, and Linear SVM without (in light grey) and considering (in dark grey) the intersection of the confidence intervals, respectively.

Now, we analyze if the results of the different strategies are statistically significant. To check for the statistically significant methods, we use the Nemenyi test - two techniques are significantly different if the corresponding average rankings differ by at least the critical difference value (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6J}} \tag{4.28}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$, k is the number of methods in the comparative, and J is the total number of performed experiments. In our case, when comparing 13 methods with a confidence value $\alpha = 0.05$, $q_{0.05} = 1.771$. Substituting in eq.(4.28), we obtain a critical difference value of 0.9955. Looking at the rankings of each decoding strategy shown in the first and third column of each group in fig. 4.6, one can observe that any variant of the Loss-Weighted strategy has a difference superior to the critical value of Type 0 and Type I strategies, only intersecting with the Laplacian and $\beta - DEN$ Type III strategies. Thus, we can argue that the LW variants are significantly better than Type 0 and Type I strategies at 95% of the confidence interval in the present experiments. In the case of the Type 0 and Type I strategies, although Type I strategies tend to have inferior ranking (thus, better position) than the Type 0 methods, there are combinations of methods for which the difference is inferior to the critical value, and therefore, we can not argue

that in those cases the Type I strategies are significantly better than the methods of Type 0.

Finally, the mean ranking positions grouping the techniques in their respective types are shown in table 4.5. One can observe that the ranking performance in all cases is better when satisfying the decoding properties, as claimed in the previous section. Besides, the novel Type III strategies obtain results statistically significantly better than the rest of the state-of-the-art strategies.

Table 4.5: Ranking positions of the decoding strategies on the UCI experiments grouped by type.

	Gentle Adaboost			Linear SVM		
	Type 0	Type I	Type III	Type 0	Type I	Type III
Discrete	5.5000	4.9844	3.3715	5.6042	5.3880	3.2951
Continuous	3.0799	2.7839	**1.7813**	3.2778	3.0469	**1.8681**

An interesting point of the previous experiments is to focus on the cases where the new decoding rules obtain the highest performance improvements. We showed that the new methods encoding the presented decoding rules improve the rest of decoding strategies for all coding designs. Looking at the tables of Appendix I, one can observe that the improvements are more significant when the new rules are applied over coding designs with higher sparseness degree (high percentage of zero symbols in the coding matrix M). It is produced because when we increase the percentage of zero symbols, the two biases produced by the third symbol also increase, and the classification performance for the traditional decoding strategies is more affected. A particular case where this effect is less significant is in the one-versus-one design, where though the sparseness is high the results for the different decoding strategies do not significantly differ. It can be explained because the amount of positions containing the zero symbol and the $\{-1, +1\}$ values coincide for all codewords, and thus, the *bias* and *dynamic range* are the same.

4.8 Decoding discussion

In this chapter, we analyzed the decoding step of the ternary symbol-based ECOC framework. We showed some inconsistences introduced by the traditional decoding strategies when using the zero symbol. Two working hypotheses to deal with a successful decoding were formulated and analyzed on a new taxonomy of decoding strategies. As a consequence, different strategies fulfilling the presented properties were proposed. The validation of the methodology was performed over a wide set of the UCI Machine Learning Repository data sets using the state-of-the-art coding and decoding strategies as well as Adaboost and Support Vector Machines as the base classifiers. We showed that when the decoding strategies avoid the *bias* introduced by the zero symbol and all the codewords work in the same *dynamic range*, significant performance improvements are obtained in the ECOC evaluation.

Four alternatives to decode were proposed. The Attenuated Euclidean decoding is useful to avoid the influence of the ECOC coding matrix positions that do not provide relevant information of the data. This is the simplest choice for ternary decoding. The strategy has a low complexity, but its performance decreases when the sparseness degree between rows of the coding matrix increases.

The Laplacian decoding introduces a measure that counts the number of coincidences between the input codeword and the class codeword, normalizing by the total number of codeword positions. The complexity of this strategy is higher than the one provided by the Attenuated Euclidean decoding, but it is tolerant to high sparseness degrees.

The Pessimistic β-density decoding slightly increase the performance and complexity of the previous Laplacian decoding but it extends the discrete behavior to continuous by estimating the probability density functions between two codewords.

Finally, the Loss-Weighted decoding has the highest estimation complexity, but it is the most suitable choice when our objective is to maximize the classification performance. The accurate results are obtained by means of a combination of probabilities that avoids the influence of the positions that do not provided information at the coding step while making the decoding measures between codewords comparable either in the binary as in the ternary ECOC framework.

Chapter 5

Separability of Ternary Codes for Sparse Designs

In the previous sections, we presented ternary ECOC designs that are better adapted to the distribution of the data than traditional approaches, and redefined the decoding step in the ternary ECOC framework. However, some coding techniques still contain inconsistences. As a result, problematic coding designs are used, and therefore, they require to be reconsidered. In this section, we present a new formulation of the ternary ECOC distance and the error-correcting capabilities in the ternary ECOC framework. Based on the new measure, we stress on how to design coding matrices preventing coding ambiguity and proposing a new Sparse Random coding matrix with ternary distance maximization.

5.1 Random ECOC Designs

In this section, we overview both Dense and Sparse Random ECOC designs [5]. We show the inconsistence of the classical Sparse Random design and introduce a new measure for sparse coding designs.

5.1.1 Dense Random Design

Given a binary ECOC matrix $M \in \{-1, 1\}^{N \times n}$, where N is the number of classes and n the codeword length, the minimum Hamming distance d_r among all pairs of rows is defined as follows [5]:

$$d_r = min \left\{ \sum_{j=1}^{n} (1 - sign(y_{i_1}^j \cdot y_{i_2}^j))/2 \right\} \tag{5.1}$$

for $i_1, i_2 \in \{1, ..., N\}$, $i_1 \neq i_2$, being $y_{i_1}^j$ the j^{th} position of the codeword for class c_{i_1}. Suppose that two codewords coded using $\{-1, +1\}$ values have a Hamming distance of three. Then, it means that even if we fail in a bit, we still are able to obtain the correct classification. It suggests that a distance d_r in a binary ECOC matrix M can correct $[d_r - 1]/2$ codeword errors at the decoding step [24]. Because of these binary error-correction capabilities, many ECOC designs, such as random ECOC strategies, base the design of the ECOC coding matrix on maximizing the value d_r [5].

Let us consider the distance d_c between all pairs of columns and their opposites:

$$d_c = min_{j_1, j_2} \{min\,(A(j_1, j_2), B(j_1, j_2))\} \tag{5.2}$$

being:

$$A(j_1, j_2) = \sum_{i=1}^{N} (1 - sign(y_i^{j_1} \cdot y_i^{j_2}))/2 \tag{5.3}$$

$$B(j_1, j_2) = \sum_{i=1}^{N} (1 - sign(-1 \cdot (y_i^{j_1}) \cdot y_i^{j_2}))/2 \tag{5.4}$$

where $j_1, j_2 \in \{1, ..., n\}$, $j_1 \neq j_2$. High value of d_c contributes to consider different sub-partitions of classes and to increase the variability of the knowledge of the classifiers. Note that in eq.(5.4) the factor (-1) is used to take into account the independence of the class ordering, i.e. the base classifier learns the same problem from the partition C_1 versus C_2 and from C_2 versus C_1.

The Dense Random ECOC strategy [5] tries to maximize simultaneously both previous d_r and d_c distances to design matrices where the decoding strategies are able to obtain a correct classification still when there exist failures in some bits of the tested codewords. The Dense random strategy generates a high number of random coding matrices M of length n, where the values $\{+1, -1\}$ have a certain probability to appear (usually $P(1) = P(-1) = 0.5$). Studies on the performance of the dense random strategy suggests a length of $n = 10 \log N$ [5]. In order to assure optimal

performance of ECOC classification, for the set of generated dense random matrices, the optimal one should maximize the Hamming Decoding measure between rows d_r and columns d_c (also considering the opposites), taking into account that each column of the matrix M must contain both different symbols $\{-1, +1\}$.

In fig. 5.1 some coding errors are shown. Fig. 5.1(a) has a dichotomizer (h_3) with all the elements coded by -1. In this case, we do not have two groups of classes to split. Fig. 5.1(b) has the hypotheses h_1 and h_4 splitting the same sub-groups of classes in opposite order, which exactly learns the same problem. The coding matrix M of fig. 5.1(c) is not able to distinguish between classes c_1 and c_3 since their respective codewords y_1 and y_3 are the same. The three previous problems in the ECOC designs do not occur when we use standard coding strategies such as one-versus-one or one-versus-all. When we use the dense random strategy, by definition [5] one needs to consider each dichotomizer to have positions coded by +1 and -1 in order to maximize the Hamming Decoding measure among the columns and their opposites; and to have a high Hamming Decoding value between rows, which prevents the errors produced in fig. 5.1(a), fig. 5.1(b), and fig. 5.1(c), respectively.

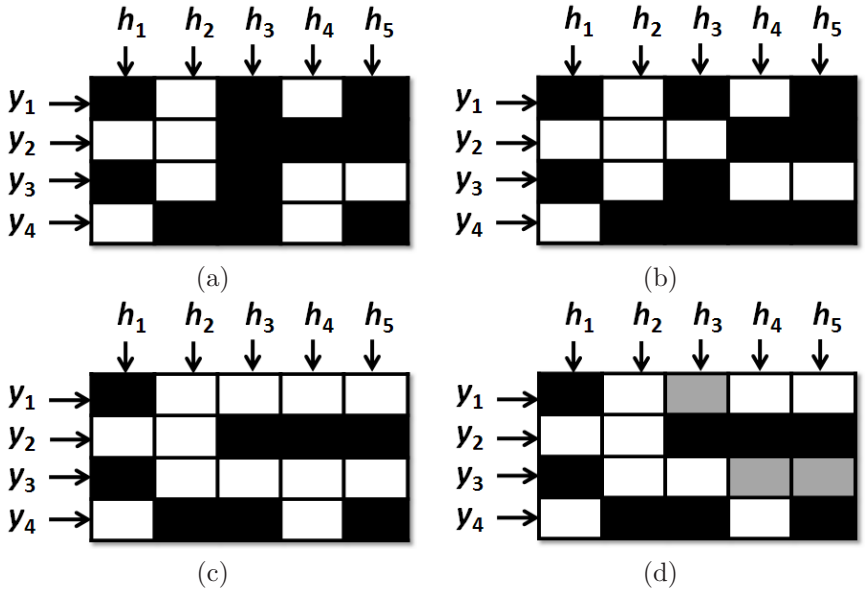


Figure 5.1: Wrong binary and ternary ECOC designs. (a) Wrong hypothesis h_3. (b) Redundant hypotheses h_1 and h_4. (c) Repeated codewords y_1 and y_3 for classes c_1 and c_3. (d) Codification error between classes c_1 and c_3.

5.1.2 Classical Sparse Random Design

One of the main limitations of the binary ECOC framework is the need of considering all classes for each binary classifier. Although a high distance d_r and d_c can be computed, the selection of the most relevant sub-partition of classes for different

multi-classification problems is not assured in the coding design. This fact implies the need of designing large codes to increase the discriminating ability of the combined set of binary problems. Moreover, taking into account the whole set of classes for each classifier significantly reduces the number of possible sub-partitions of classes to consider.

To take into account a higher number of possible classifiers, a third symbol was introduced in the ECOC framework [5]. In this sense, the Sparse Random strategy is designed in the same way than the Dense design, but it includes the third symbol zero with another probability to appear, given by $P(0) = 1 - P(-1) - P(1)$. Studies suggest a sparse code length of $15 \log N$ [5].

We consider that to increase the class separability in the ternary ECOC framework, the distance d_c of the binary case can be maintained since all three symbols $\{-1, 0, +1\}$ have influence on the information learnt by each dichotomizer. It means that the distance between columns produced by the positions coded by zero increases the variability of the classifiers. However, we argue that the use of the codewords separability maximizing the measure d_r to design a Sparse Random matrix may contain inconsistency.

5.1.3 Sparse Random Design with Ternary Separability

Let us show an example to analyze sparse designs. A zero symbol in a class code introduces *one degree of freedom*, that means that both +1 and -1 are possible values during the test classification since the class has not been taken into account to train the corresponding dichotomizer. Any codeword y_i containing the zero symbol defines an extended set of possible codewords that could be obtained by examples of the class c_i. In this sense, a possible codeword $y_1 = \{1, 0, 0\}$ can be disambiguated into its extended set of codewords $Y_1^e = \{\{1, 1, 1\}, \{1, 1, -1\}, \{1, -1, 1\}, \{1, -1, -1\}\}$, where each of the four codewords of y_1 is a possible representation[1] of the same codeword y_1. Now, a possible codeword for a second class $y_2 = \{1, 1, 1\}$ corresponds to one of the four possible representations of y_1 ($y_2 \in Y_1^e$).

Let us consider another example of codewords of length six. Suppose that we randomly define two codewords $y_1 = \{1, 1, 1, 0, 0, 0\}$ and $y_2 = \{0, 0, 0, 1, 1, 1\}$ in a Sparse Random design. If we use the classical distance d_r between y_1 and y_2, we obtain a class separability of three. However, based on the previous example, if we disambiguate y_1 and y_2, we obtain that $Y_1^e \cap Y_2^e = \{1, 1, 1, 1, 1, 1\}$. Thus, an input test codeword $X = \{1, 1, 1, 1, 1, 1\}$ belongs to both previous codewords, which implies a wrong Sparse design.

Finally, observe the ternary coding matrix M of fig. 5.1(d). Suppose that the matrix M of the figure receives an input test data sample which codeword corresponds to $X = \{-1, 1, 1, 1, 1\}$. This codeword matches with the four positions different of zero from class c_1 and the three from class c_3. In this case, $X \in Y_1^e$ and $X \in Y_3^e$. Thus, both classes can be a possible solution. However, the HD between codewords y_1 and y_3 produces a value of 1.5. Note that in the literature [5], a Sparse Random matrix is generated by selecting the matrix from a previous set of matrices that maximizes the distances d_r and d_c. As commented, the HD between columns containing the

[1] Possible representation means that any test example of class c_1 would give a codeword from Y_1^e.

third symbol is still useful since the zero positions help to create create a rich set of partitions to be learnt. However, the measure d_r for the row separability in terms of the HD, as claimed, is inconsistent. Instead, to assure that the coding matrix M splits all pairs of classes, each pair of codewords of M should be split by at least one hypothesis:

Definition 8.: The **ternary separability** condition of a matrix M is defined as:

$$\forall(y_{i_1}, y_{i_2})|i_1, i_2 \in \{1, ..., N\}, i_1 \neq i_2, \exists h_j|(c_{i_1} \in C_1^j, c_{i_2} \in C_2^j) \vee (c_{i_2} \in C_1^j, c_{i_1} \in C_2^j) \tag{5.5}$$

where C_1^j and C_2^j are the two subsets of classes for hypothesis h_j, respectively. Then, we can define the distance between two codewords in a ternary symbol-based ECOC:

Definition 9.: The **ternary distance** between two codewords (y_1,y_2) is defined as:

$$d(y_1, y_2) = \sum_{j=1}^{n} \frac{1}{2}|y_1^j||y_2^j|(1 - y_1 y_2) \tag{5.6}$$

It defines the number of different bits between two codewords without taking into account the positions coded by zero. Note that the term $\frac{1}{2}(1 - y_1 y_2)$ is equivalent to the standard Hamming distance estimated in the binary case expressed in a more compact way. Thus, the weighting term $|y_1^j||y_2^j|$ makes the distance to ignore the zero positions which do not give information about the classes separability. Then, the pair of codewords (y_{i_1}, y_{i_2}) that are split by the minimum number of hypothesis in a ternary ECOC matrix M defines the new distance d_t:

Definition 10.: The **distance** d_t of a coding matrix M is defined as follows:

$$d_t = argmin_{i_1, i_2} \sum_{j=1}^{n} \frac{1}{2}|y_{i_1}^j||y_{i_2}^j|(1 - y_{i_1} y_{i_2}) \tag{5.7}$$

where the term d_t defines the distance between the pair of codewords that are split by the minimum number of binary problems in a ternary symbol-based ECOC matrix.

Based on the new ternary distance, we can define the error-correcting capabilities in the ternary ECOC framework. As the distance in the ternary case has been reformulated, the new measure of error-correction also changes. Having a N-multi-class classification problem in the binary ECOC framework, a distance d_r between rows of M can correct $[d_r - 1]/2$ bits errors. In the ternary case, the maximum class separability is defined by the measure d_t. Thus, on a sparse ECOC matrix, $[d_t - 1]/2$ bits errors can be corrected[2].

As the use of the distance d_r applied to the classical design of the Sparse Random matrix M produces inconsistences, we suggest to redefine the coding stage of the Sparse Random designs. A good codification of a ternary matrix should assure the highest number of codeword bits splitting each pair of rows; that is to maximize the

[2]We realize that the error-correcting capability also depends on the way that the decoding strategies are applied.

value d_t. Therefore, we propose to use the new measure of ternary separability for the Sparse Random design. In this case, the selected random matrix should be that one which maximizes simultaneously d_c and d_t.

5.2 Sparse Design Evaluation

We discuss the data, comparatives, and measurements of the experiments before the results are presented[3].

• *Data*: The data used for the experiments consists of 16 multi-class data sets from the UCI Machine Learning Repository data set [8]. The details of the data sets can be found in chapter F.

• *Comparatives*: For the comparatives, we use the classical Sparse Random design [5] and the new Sparse Random with ternary distance maximization. The sparse matrices are selected from a set of 20000 randomly generated matrices with a length of codewords of N. To decode, we use the state-of-the-art decoding strategies and the new decoding designs presented on previous chapters: Hamming Decoding, Euclidean Decoding, Inverse Hamming Decoding, Attenuated Euclidean Decoding, Loss-based Decoding with Linear and Exponential Loss-functions, Probabilistic Decoding, Laplacian Decoding, Pessimistic β-Density Distribution Decoding, Linear Loss-Weighted with discrete and continuous outputs of the classifier, and the Exponential Loss-Weighted with discrete and continuous outputs of the classifier.

• *Measurements*: To measure the performance of the different strategies we apply stratified ten-fold cross-validation and test for confidence interval at 95% with a two-tailed t-test. The base classifiers used for the experiments are Gentle Adaboost with 50 runs of decision stumps and the Linear Support Vector Machines.

UCI classification

In this experiment, we classify the 16 multi-class UCI Machine Learning Repository data sets. To test the Sparse Random strategies, we generated a set of 20000 arbitrary random matrices, where the probabilities of appearance of each symbol are $P(0) = P(1) = P(-1) = 1/3$. From exactly the same set of generated matrices, we selected the classical Sparse Random matrix by the one which maximizes d_r and d_c, and the new Sparse Random matrix by selecting the one which maximizes d_t and d_c. To decode, the commented 13 decoding strategies are applied over the Sparse Random designs for Gentle Adaboost and Linear SVM as the base classifiers.

Tables 5.2 and 5.3 show the performance results on the UCI data sets for the Sparse Random designs using Gentle Adaboost and Linear SVM, respectively. The nomenclature used for the data sets is shown in the table 5.1. For each data set shown in the tables 5.2 and 5.3, the results on the top correspond to the performance and confidence interval using the classical Sparse Random strategy. The results on the bottom correspond to the results using the Sparse Random selection based on maximizing the new ternary distance. Note that in most cases, the new Sparse design outperform the results of the classical one. Only in few cases, such as at the Satimage data set with SVM or the Iris data set with Adaboost, there are some performances inferior to the classical approach.

To show the performance improvements by selecting the new Sparse Random matrix, the absolute and relative improvements are shown in fig. 5.2 for Gentle Adaboost and Linear SVM, respectively. The relative improvement is computed as the division between the performance of the new Sparse design and the classical one, and the

[3]More experimental results and analysis of the Sparse design methodology are shown in the Applications chapter.

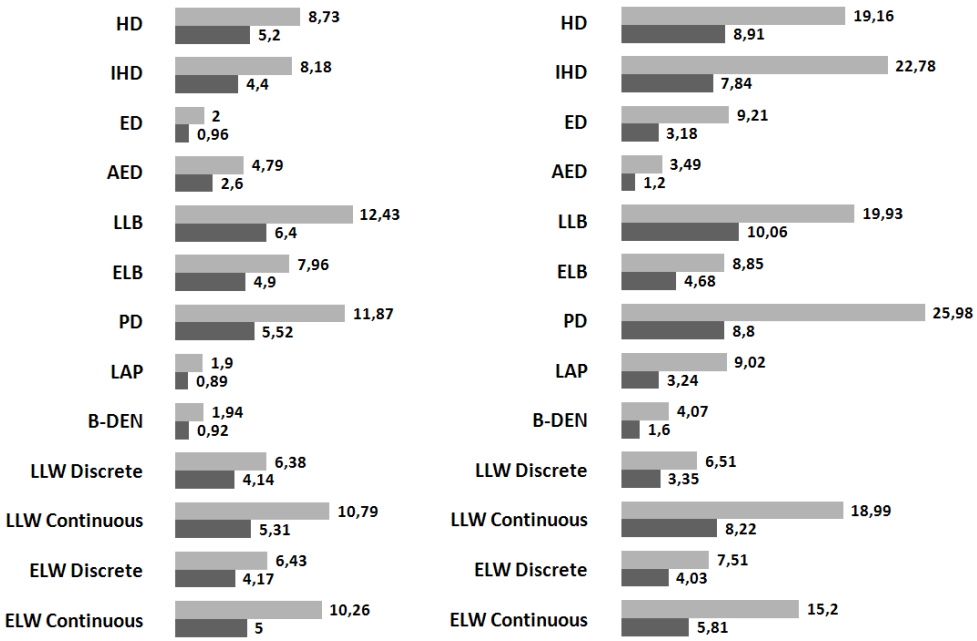


Figure 5.2: Absolute (light lines) and relative (dark lines) improvement for the Sparse Random designs using ternary distance maximization for Gentle Adaboost (left) and Linear SVM (right) on the UCI experiments, respectively.

absolute improvement correspond to the direct difference of performances. The light bars correspond to the relative improvement, and the dark lines to the absolute one. Note that simply changing the decision on the selection of the sparse matrix from the same set of generated random matrices, the performance significantly increases independently of the decoding strategy applied. It is produced since the maximization of d_t assures us to select the matrix with the higher number of bits splitting codewords (and thus, classes).

Table 5.1: Codification of the UCI data sets.

A	Dermathology	I	Yeast
B	Iris	J	Satimage
C	Ecoli	K	Letter
D	Wine	L	Pendigits
E	Glass	M	Segmentation
F	Thyroid	N	OptDigits
G	Vowel	O	Shuttle
H	Balance	P	Vehicle

As a conclusion of the experiments, we can state that the ternary distance d_t based

Table 5.2: Sparse Random results using Gentle Adaboost on the UCI data sets.

	HD	*IHD*	*ED*	*AED*	*LLB*	*ELB*	*PD*	*LAP*	*βDEN*	*LLW* Disc.	*LLW* Cont.	*ELW* Disc.	*ELW* Cont.
A	0.588	0.634	0.636	0.647	0.549	0.587	0.444	0.636	0.636	0.650	0.452	0.650	0.436
	0.027	0.012	0.011	0.009	0.017	0.021	0.035	0.011	0.011	0.008	0.051	0.008	0.042
	0.926	0.923	0.926	0.923	0.896	0.926	0.945	0.926	0.926	0.929	0.920	0.929	0.940
	0.017	0.018	0.017	0.015	0.024	0.021	0.013	0.017	0.017	0.017	0.015	0.017	0.015
B	0.933	0.933	0.933	0.933	0.953	0.953	0.953	0.933	0.933	0.933	0.953	0.933	0.953
	0.019	0.019	0.019	0.019	0.014	0.014	0.014	0.019	0.019	0.019	0.014	0.019	0.014
	0.926	0.926	0.926	0.926	0.960	0.960	0.960	0.926	0.926	0.933	0.960	0.933	0.960
	0.020	0.020	0.020	0.020	0.014	0.014	0.014	0.020	0.020	0.019	0.014	0.019	0.014
C	0.373	0.379	0.367	0.533	0.284	0.302	0.493	0.370	0.357	0.539	0.443	0.551	0.477
	0.017	0.016	0.021	0.014	0.019	0.020	0.017	0.018	0.021	0.015	0.033	0.011	0.029
	0.373	0.379	0.367	0.533	0.284	0.302	0.493	0.370	0.357	0.539	0.443	0.551	0.477
	0.017	0.016	0.021	0.014	0.019	0.020	0.017	0.018	0.021	0.015	0.033	0.011	0.029
D	0.943	0.943	0.943	0.943	0.960	0.954	0.954	0.943	0.943	0.943	0.960	0.943	0.960
	0.018	0.018	0.018	0.018	0.011	0.013	0.013	0.018	0.018	0.018	0.011	0.018	0.011
	0.949	0.949	0.949	0.949	0.960	0.960	0.954	0.949	0.949	0.949	0.954	0.949	0.960
	0.012	0.012	0.012	0.012	0.011	0.011	0.013	0.012	0.012	0.012	0.013	0.012	0.011
E	0.592	0.569	0.592	0.588	0.486	0.598	0.614	0.592	0.592	0.592	0.530	0.592	0.605
	0.037	0.040	0.037	0.036	0.036	0.039	0.032	0.037	0.037	0.037	0.028	0.037	0.036
	0.655	0.646	0.645	0.645	0.626	0.640	0.643	0.645	0.645	0.645	0.579	0.645	0.625
	0.026	0.025	0.032	0.033	0.031	0.028	0.034	0.032	0.032	0.032	0.035	0.032	0.032
F	0.907	0.907	0.907	0.907	0.921	0.921	0.911	0.907	0.907	0.907	0.921	0.907	0.921
	0.026	0.026	0.026	0.026	0.027	0.027	0.026	0.026	0.026	0.026	0.027	0.026	0.027
	0.898	0.898	0.898	0.898	0.921	0.921	0.911	0.898	0.898	0.898	0.921	0.898	0.921
	0.025	0.025	0.025	0.025	0.027	0.027	0.026	0.025	0.025	0.025	0.027	0.025	0.027
G	0.382	0.279	0.441	0.430	0.362	0.396	0.439	0.443	0.443	0.451	0.405	0.449	0.431
	0.020	0.012	0.022	0.025	0.023	0.024	0.020	0.021	0.021	0.024	0.022	0.025	0.019
	0.443	0.373	0.452	0.441	0.449	0.465	0.452	0.454	0.454	0.441	0.472	0.441	0.481
	0.023	0.021	0.025	0.023	0.031	0.029	0.023	0.026	0.026	0.024	0.027	0.024	0.027
H	0.425	0.425	0.801	0.801	0.481	0.639	0.785	0.801	0.801	0.788	0.710	0.788	0.735
	0.020	0.020	0.040	0.040	0.027	0.048	0.037	0.040	0.040	0.051	0.050	0.051	0.052
	0.504	0.504	0.504	0.504	0.730	0.721	0.800	0.504	0.504	0.809	0.756	0.809	0.756
	0.061	0.061	0.061	0.061	0.079	0.079	0.077	0.061	0.061	0.082	0.077	0.082	0.077
I	0.433	0.402	0.435	0.393	0.421	0.415	0.345	0.435	0.435	0.395	0.401	0.402	0.403
	0.021	0.011	0.020	0.008	0.014	0.015	0.008	0.019	0.019	0.013	0.020	0.016	0.018
	0.435	0.408	0.436	0.454	0.429	0.425	0.464	0.435	0.435	0.447	0.413	0.447	0.425
	0.012	0.011	0.012	0.013	0.014	0.013	0.010	0.012	0.012	0.014	0.015	0.014	0.014
J	0.795	0.766	0.814	0.639	0.766	0.787	0.637	0.814	0.814	0.673	0.638	0.656	0.705
	0.019	0.018	0.018	0.014	0.025	0.026	0.021	0.018	0.018	0.025	0.034	0.015	0.030
	0.789	0.776	0.814	0.807	0.814	0.820	0.829	0.814	0.814	0.818	0.832	0.818	0.833
	0.019	0.017	0.021	0.020	0.019	0.019	0.017	0.021	0.021	0.019	0.020	0.019	0.020
K	0.803	0.821	0.823	0.841	0.821	0.828	0.812	0.834	0.838	0.848	0.855	0.862	0.880
	0.016	0.016	0.018	0.018	0.017	0.017	0.017	0.017	0.015	0.014	0.015	0.015	0.016
	0.839	0.840	0.850	0.863	0.836	0.845	0.834	0.860	0.876	0.872	0.885	0.874	0.889
	0.016	0.018	0.016	0.017	0.017	0.017	0.016	0.017	0.016	0.014	0.015	0.014	0.015
L	0.839	0.818	0.858	0.889	0.836	0.857	0.848	0.927	0.932	0.932	0.940	0.932	0.947
	0.011	0.010	0.009	0.010	0.009	0.007	0.011	0.010	0.010	0.008	0.009	0.007	0.010
	0.859	0.848	0.883	0.921	0.872	0.889	0.869	0.942	0.942	0.952	0.953	0.960	0.967
	0.010	0.010	0.010	0.010	0.009	0.007	0.011	0.009	0.011	0.007	0.008	0.006	0.011
M	0.921	0.922	0.921	0.891	0.711	0.863	0.920	0.921	0.921	0.927	0.865	0.928	0.925
	0.010	0.009	0.010	0.012	0.016	0.006	0.010	0.010	0.010	0.010	0.008	0.009	0.008
	0.939	0.933	0.938	0.933	0.897	0.919	0.938	0.939	0.938	0.938	0.935	0.938	0.941
	0.009	0.010	0.009	0.009	0.015	0.014	0.008	0.009	0.009	0.009	0.009	0.009	0.009
N	0.753	0.716	0.796	0.740	0.787	0.795	0.783	0.795	0.796	0.769	0.794	0.773	0.809
	0.018	0.016	0.022	0.023	0.026	0.025	0.024	0.023	0.022	0.021	0.025	0.021	0.020
	0.769	0.651	0.811	0.779	0.685	0.724	0.810	0.811	0.811	0.815	0.772	0.815	0.803
	0.022	0.016	0.025	0.030	0.018	0.019	0.023	0.026	0.026	0.023	0.025	0.023	0.024
O	0.658	0.703	0.702	0.710	0.653	0.669	0.716	0.702	0.702	0.702	0.691	0.702	0.699
	0.023	0.024	0.023	0.036	0.021	0.022	0.027	0.023	0.023	0.023	0.019	0.023	0.019
	0.723	0.724	0.723	0.724	0.730	0.734	0.727	0.723	0.723	0.727	0.730	0.729	0.730
	0.033	0.029	0.033	0.032	0.031	0.029	0.025	0.033	0.033	0.033	0.034	0.032	0.030
P	0.850	0.849	0.998	0.998	0.854	0.859	0.944	0.998	0.998	0.998	0.936	0.998	0.990
	0.000	0.000	0.000	0.000	0.002	0.004	0.008	0.000	0.000	0.000	0.022	0.000	0.006
	0.998	0.989	0.998	0.998	0.779	0.957	0.853	0.998	0.998	0.998	0.817	0.998	0.967
	0.000	0.003	0.000	0.000	0.067	0.020	0.152	0.000	0.000	0.000	0.078	0.000	0.021

on maximizing the ternary separability allows high splitting of the classes codewords. In the previous experiments significant performance improvements are obtained, independently of the decoding strategy applied, when the sparse matrix is selected by maximizing the d_t criterion. Note that the classical sparse matrix is selected from the same set of matrices as the new sparse matrix, but it obtains very inferior results. This suggests that for designs that consider the new measures, class separability is increased. Thus, the decoding strategies are able to distinguish among different codewords with higher confidence. Moreover, the ternary distance can be applied to problem-dependent ECOC schemes, assuring the consistence of the designs. At the

Table 5.3: Sparse Random results using Linear SVM on the UCI data sets.

	HD	IHD	ED	AED	LLB	ELB	PD	LAP	βDEN	LLW Disc.	LLW Cont.	ELW Disc.	ELW Cont.
A	0.374	0.382	0.440	0.868	0.456	0.623	0.853	0.440	0.719	0.870	0.766	0.870	0.835
	0.005	0.007	0.024	0.017	0.042	0.027	0.030	0.024	0.024	0.015	0.060	0.015	0.041
	0.936	0.847	0.936	0.939	0.950	0.950	0.961	0.936	0.936	0.936	0.953	0.933	0.953
	0.011	0.028	0.011	0.011	0.010	0.010	0.009	0.011	0.011	0.011	0.010	0.012	0.010
B	0.666	0.666	0.973	0.973	0.666	0.920	0.773	0.973	0.973	0.973	0.780	0.973	0.933
	0.010	0.010	0.010	0.010	0.010	0.019	0.019	0.010	0.010	0.010	0.023	0.010	0.016
	0.720	0.720	0.720	0.720	0.926	0.926	0.826	0.720	0.720	0.973	0.940	0.973	0.933
	0.025	0.025	0.025	0.025	0.020	0.020	0.022	0.025	0.025	0.010	0.020	0.010	0.021
C	0.684	0.269	0.726	0.723	0.699	0.743	0.275	0.726	0.726	0.683	0.413	0.613	0.411
	0.022	0.025	0.031	0.030	0.026	0.031	0.029	0.031	0.031	0.038	0.033	0.058	0.042
	0.758	0.726	0.758	0.737	0.770	0.761	0.766	0.758	0.758	0.667	0.616	0.711	0.610
	0.026	0.029	0.026	0.023	0.029	0.027	0.031	0.026	0.026	0.037	0.044	0.026	0.047
D	0.932	0.932	0.932	0.932	0.955	0.955	0.949	0.932	0.932	0.932	0.955	0.932	0.955
	0.016	0.016	0.016	0.016	0.013	0.013	0.009	0.016	0.016	0.016	0.013	0.016	0.013
	0.932	0.932	0.932	0.932	0.955	0.955	0.949	0.932	0.932	0.932	0.955	0.932	0.955
	0.016	0.016	0.016	0.016	0.013	0.013	0.009	0.016	0.016	0.016	0.013	0.016	0.013
E	0.438	0.460	0.446	0.457	0.428	0.456	0.452	0.452	0.446	0.443	0.457	0.427	0.458
	0.024	0.029	0.033	0.023	0.025	0.029	0.022	0.028	0.033	0.021	0.030	0.031	0.033
	0.503	0.509	0.503	0.484	0.552	0.523	0.524	0.503	0.503	0.504	0.517	0.504	0.534
	0.028	0.019	0.028	0.029	0.028	0.030	0.028	0.028	0.028	0.043	0.031	0.043	0.036
F	0.814	0.814	0.943	0.943	0.818	0.948	0.897	0.943	0.943	0.943	0.856	0.943	0.948
	0.009	0.009	0.021	0.021	0.007	0.015	0.023	0.021	0.021	0.021	0.011	0.021	0.020
	0.916	0.916	0.916	0.916	0.934	0.934	0.855	0.916	0.916	0.943	0.934	0.943	0.934
	0.026	0.026	0.026	0.026	0.025	0.025	0.019	0.026	0.026	0.021	0.025	0.021	0.025
G	0.343	0.308	0.341	0.301	0.313	0.298	0.241	0.352	0.352	0.357	0.328	0.359	0.360
	0.018	0.018	0.017	0.021	0.021	0.020	0.025	0.018	0.018	0.025	0.029	0.025	0.023
	0.382	0.376	0.366	0.314	0.365	0.367	0.269	0.360	0.362	0.368	0.334	0.364	0.379
	0.026	0.029	0.018	0.010	0.027	0.022	0.018	0.016	0.017	0.018	0.021	0.020	0.015
H	0.855	0.855	0.855	0.855	0.833	0.833	0.822	0.855	0.855	0.855	0.845	0.855	0.855
	0.041	0.041	0.041	0.041	0.035	0.035	0.040	0.041	0.041	0.041	0.045	0.041	0.041
	0.855	0.855	0.855	0.855	0.833	0.833	0.822	0.855	0.855	0.855	0.845	0.855	0.855
	0.041	0.041	0.041	0.041	0.035	0.035	0.040	0.041	0.041	0.041	0.045	0.041	0.041
I	0.380	0.385	0.380	0.378	0.379	0.390	0.217	0.380	0.381	0.341	0.210	0.346	0.221
	0.012	0.013	0.012	0.013	0.016	0.012	0.005	0.012	0.013	0.017	0.009	0.021	0.006
	0.491	0.476	0.493	0.489	0.492	0.495	0.506	0.493	0.493	0.484	0.472	0.483	0.497
	0.018	0.015	0.019	0.018	0.014	0.019	0.016	0.018	0.018	0.024	0.024	0.024	0.031
J	0.718	0.710	0.726	0.615	0.670	0.725	0.634	0.726	0.726	0.618	0.403	0.638	0.660
	0.016	0.014	0.014	0.019	0.048	0.019	0.021	0.014	0.014	0.019	0.028	0.030	0.021
	0.724	0.626	0.734	0.732	0.656	0.750	0.739	0.734	0.734	0.776	0.489	0.776	0.782
	0.014	0.009	0.013	0.014	0.014	0.011	0.012	0.013	0.013	0.017	0.037	0.017	0.018
K	0.632	0.637	0.648	0.662	0.648	0.652	0.643	0.671	0.672	0.702	0.705	0.703	0.710
	0.010	0.009	0.008	0.009	0.014	0.009	0.010	0.011	0.009	0.010	0.008	0.009	0.009
	0.642	0.653	0.663	0.675	0.649	0.660	0.648	0.678	0.692	0.708	0.717	0.718	0.729
	0.010	0.009	0.007	0.008	0.006	0.007	0.008	0.010	0.008	0.007	0.008	0.008	0.010
L	0.878	0.887	0.893	0.912	0.902	0.902	0.897	0.913	0.916	0.917	0.921	0.918	0.927
	0.008	0.007	0.008	0.009	0.009	0.010	0.009	0.008	0.009	0.008	0.014	0.013	0.013
	0.897	0.908	0.917	0.932	0.910	0.912	0.901	0.938	0.939	0.941	0.944	0.948	0.953
	0.010	0.009	0.008	0.009	0.014	0.013	0.011	0.010	0.009	0.008	0.014	0.009	0.010
M	0.706	0.627	0.838	0.800	0.475	0.849	0.716	0.837	0.837	0.837	0.700	0.837	0.851
	0.013	0.015	0.006	0.005	0.010	0.007	0.003	0.007	0.007	0.007	0.030	0.007	0.007
	0.793	0.727	0.800	0.810	0.727	0.843	0.751	0.840	0.840	0.844	0.791	0.844	0.856
	0.005	0.012	0.005	0.006	0.010	0.007	0.006	0.005	0.005	0.006	0.014	0.006	0.007
N	0.710	0.664	0.767	0.738	0.616	0.763	0.719	0.769	0.769	0.769	0.620	0.768	0.813
	0.019	0.021	0.017	0.021	0.029	0.012	0.020	0.016	0.016	0.015	0.047	0.018	0.024
	0.795	0.573	0.797	0.785	0.664	0.845	0.832	0.797	0.797	0.812	0.719	0.812	0.847
	0.030	0.016	0.029	0.027	0.021	0.030	0.025	0.029	0.029	0.031	0.023	0.031	0.030
O	0.520	0.722	0.703	0.702	0.670	0.730	0.620	0.703	0.703	0.703	0.704	0.703	0.728
	0.011	0.016	0.022	0.022	0.026	0.017	0.029	0.022	0.022	0.022	0.025	0.022	0.014
	0.728	0.728	0.728	0.730	0.742	0.781	0.763	0.728	0.728	0.736	0.751	0.736	0.776
	0.025	0.025	0.025	0.027	0.026	0.018	0.018	0.025	0.025	0.025	0.026	0.025	0.021
P	0.977	0.977	0.977	0.977	0.892	0.977	0.969	0.977	0.977	0.977	0.977	0.977	0.977
	0.003	0.003	0.003	0.003	0.014	0.003	0.007	0.003	0.003	0.003	0.003	0.003	0.003
	0.977	0.977	0.977	0.977	0.902	0.977	0.971	0.977	0.977	0.977	0.977	0.977	0.977
	0.003	0.003	0.003	0.003	0.013	0.003	0.008	0.003	0.003	0.003	0.003	0.003	0.003

same time, the new measures can also help the decoding strategies to evaluate those positions of codewords that directly affect class separability.

5.3 Separability of Sparse designs discussion

In previous section, we redefined the decoding strategies. In this section, we showed that the coding step also is affected by the zero symbol. We showed that the rows separability in terms of the Hamming distance of the binary ECOC framework can not be applied in the ternary case, and we presented a new formulation of the ternary ECOC distance and the error-correcting capabilities in the ternary ECOC framework. Based on the new measure, we stressed on how to design coding matrices preventing codification ambiguity and proposed a new Sparse Random coding matrix with ternary distance maximization.

This new formulation of Sparse random design is suitable in cases where we want to use an ECOC scheme without taking into account the information provided by the domain of the problem we are working on.

Chapter 6

Object Recognition

In object recognition, a new instance is categorized according to the pool of trained objects (cars, motorbikes, horses, flowers, etc.). As commented at previous sections, a powerful multi-class pattern and object recognition system requires to make use of a rich feature set that universally describes the data so that the new representation should minimize intra-class variability and increase inter-class variability. In order to describe objects with discriminative features, in this section, we propose two techniques for object detection and description so that the obtained features can be used in combination with the previous proposed ECOC coding and decoding designs in order to increase the categorization performance.

First, we introduce a technique that considers a region as an object, and describes its content by considering the relevant gradient magnitude points to define a probability density map of the shape of the object, even if it suffers from irregular deformations. And second, we introduce a new object detection method based on training the discriminant features of the object description. Such description includes the information of correlograms to learn at the same time the object local representation and the spatial relationship among its parts fragments.

6.1 Blurred Shape Models Descriptors

To describe an object that can suffer from irregular deformations, we propose to codify its shape by determining its external appearance. External appearance pixels use to have high gradient magnitude. Taking into account those pixels, the Blurred Shape Model descriptor defines spatial regions where some parts of the object can be involved. For this task, the input region to describe should be binary, and the activated pixels (those set to one) should belong to the shape of the object.

To process the image and obtain the object *shape*, different pre-processing techniques can be applied depending on each particular problem domain. For instance, in the case of handwriting symbols, the skeleton is a good choice since it maintains the structure of the symbols for different author strokes. In the case of grey-level or binary object classes, as the one shown in fig. 6.1(c), a contour map is more suitable to obtain the structure map.

(a) (b) (c) (d)

Figure 6.1: Object *shape* estimation by means of (a)(b) skeleton and (c)(d) Contour map.

Before applying the proposed descriptor, a *shape* alignment process is performed. This process is composed of two steps: the first step, provides invariance to rotation by means of the Hotelling transform. And the second step deals with the possible mirroring effect.

The Hotelling transform finds a new coordinate system equivalent to locating the main axis of the object. Given a set of n representative object points defined as pairs of coordinates $\mathbf{x} = (x_i, y_i)$, where $i \in [1, .., n]$, the center of mass of the object $m_{\mathbf{x}}$, and the eigenvectors V of the covariance matrix, the new transformation is obtained by means of the projection of the centered points of the object in the following way:

$$\mathbf{x}'_i = V(\mathbf{x}_i - m_{\mathbf{x}}), i \in [1, .., n] \tag{6.1}$$

Using this transform, we find the common axes for the different object instances. In fig. 6.2(a), the mean *shape* for the samples of the MPEG07 category shown in fig. 6.2(b) after applying the Hotelling transform is shown. One can observe that the *shapes* are not properly aligned. For this reason, a second step, consisting of an area density estimation process is used. Horizontal and vertical projections are applied to obtain the area of the object. Then, this area is projected on the two axes. The final alignment is obtained by horizontal and vertical reflection of the object in the direction of the higher area projections. The result of adjusting the alignment is shown in fig. 6.2(c). Another example of alignment for two MPEG07 object categories is shown in fig. 6.3.

(a) (b) (c)

Figure 6.2: (a) Mean aligned *shape* based on principal components. (b) Horizontal and vertical area estimation. (c) Readjusted alignment.

(a) (b) (c) (d)

Figure 6.3: Mean aligned *shapes* for two MPEG07 categories.

At this point, given a shape image forming the shape $S = \{x_1, .., x_m\}$, we treat each point x_i, called from now SP, as a feature to compute the BSM descriptor of the symbol shape. The image region is divided in a grid of $n \times n$ equal-sized sub-regions (cells) r_i. Each cell receives votes from the SPs in it and also from the SPs in the neighboring sub-regions. Thus, each SP contributes to a density measure of its cell and its neighboring ones, and thus, the grid size identifies the blurring level allowed for the shape. This contribution is weighted according to the distance between the point and the center of coordinates c_i of the region r_i. The algorithm is summarized in table 6.1.

In Fig. 6.4, a shape description is shown for an apple data sample. Figure 6.4(a) shows the distances d_i of a SP to the nearest sub-regions centers. To give the same importance to each SP, all the distances to the neighbor centers are normalized. The output descriptor is a vector histogram v of length $n \times n$, where each position corresponds to the spatial distribution of SPs in the context of the sub-region and their neighbors ones. Fig. 6.4(b) shows the vector descriptor updating once the distances of the first point in Fig. 6.4(a) are computed. Observe that the position of the descriptor corresponding to the affected sub-region r_{15}, which centroid is nearest to the analyzed SP, obtains a higher value.

The resulting vector histogram, obtained by processing all SPs, is normalized in the range $[0, 1]$ to obtain the probability density function (pdf) of $n \times n$ bins. In this way, the output descriptor represents a distribution of probabilities of the symbol structure considering spatial distortions, where the distortion level allowed is determined by the grid size. The BSM descriptors for different grid sizes applied to the previous example of Fig. 6.4 are shown in Fig. 6.5. Concerning the computational complexity, for a region of $n \times n$ pixels, the k relevant considered SPs to obtain the

Given an image I:

1. Obtain the *shape* S contained in I

2. Divide I in $n \times n$ equal size sub-regions $R = \{r_1, ..., r_{n^2}\}$, with c_i the center of coordinates for each region r_i.

3. Let $N(r_i)$ be the neighbor regions of region r_i, defined as $N(r_i) = \{r_k | r_k \in R, ||c_k - c_i|| \leq 2|g|\}$, where g is the cell size.

4.

For each point $\mathbf{x} \in S$,

For each $r_i \in N(r_{\mathbf{x}})$,

$d_i = d(\mathbf{x}, r_i) = ||\mathbf{x} - c_i||^2$

End_For

Update the probability vector v as:

$v(r_i) = v(r_i) + \frac{1}{d_i D_i}, \quad D_i = \sum_{c_k \in N(r_i)} \frac{1}{||\mathbf{x} - c_k||^2}$

End_For

5. Normalize the vector v as: $v = \frac{v(i)}{\sum_{j=1}^{n^2} v(j)} \forall i \in [1, ..., n^2]$

Table 6.1: BSM algorithm.

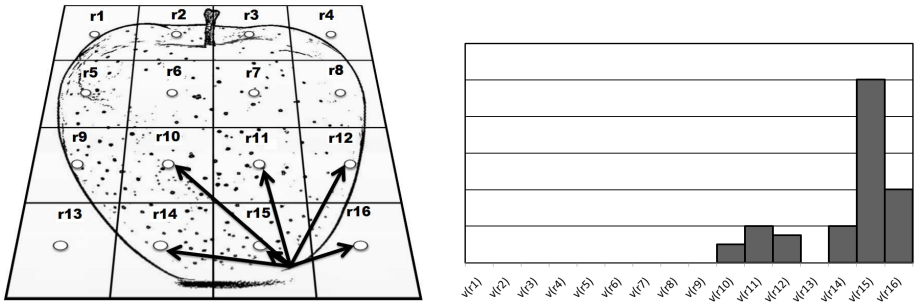


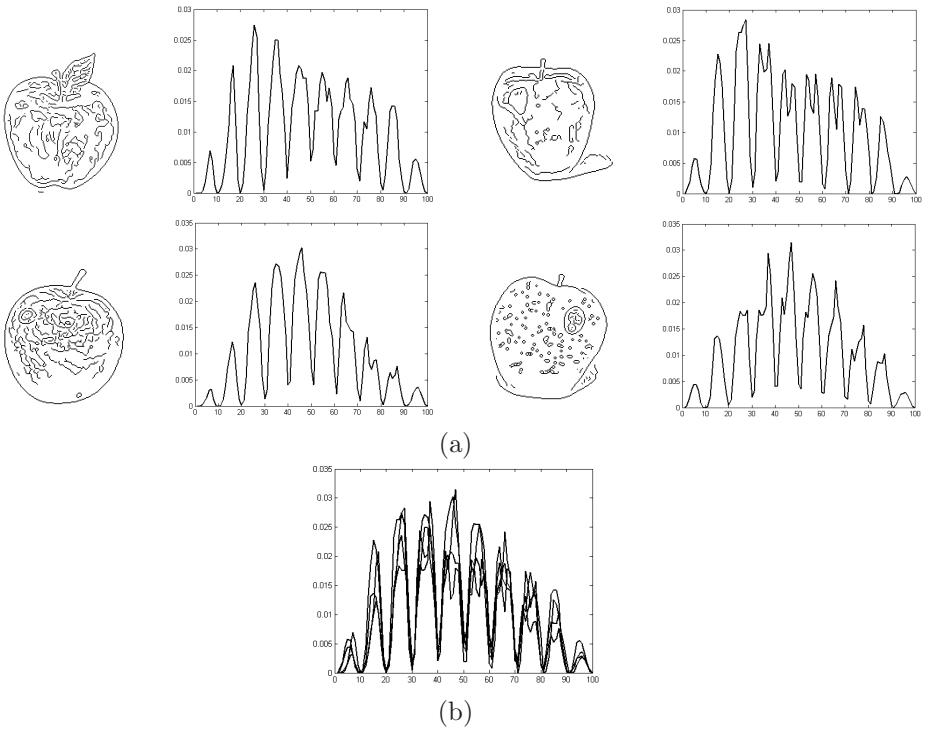
Figure 6.4: BSM density estimation example.

BSM descriptor require a cost of $O(k)$ simple operations. In Fig. 6.6(a) four BSM descriptors of apple samples of length 10×10 are shown. Figure 6.6(b) shows the correlation of the four previous descriptors. Note that though it exists some variations on the shape of the symbols, the four descriptors remain closely correlated.

An important point of the BSM description is the selection of the grid size. The optimum size defines the optimum grid encoding the blurring degree based on a particular data set distortions. Because of this reason, a common way to look for the optimum grid size is applying cross-validation over the data for different descriptor parameters. The selected grid is the one which attains the highest performance on a validation subset, defining the optimum grid encoding the different distortions over each particular problem, and offering the required tradeoff between inter-class and

(a) (b) (c) (d) (e)

Figure 6.5: (a) Input *shape*. BSM for (b) 8×8, (c) 16×16, (d) 32×32, and (e) 64×64 grid sizes.



(a)

(b)

Figure 6.6: (a) Plots of BSM descriptors of length 10×10 for four apple samples. (b) Correlation of previous BSM descriptors.

intra-class variability in a problem-dependent way.

6.2 Boosted Landmarks of Contextual Descriptors

In this section, we introduce a new object detection method based on training the discriminant features of the object description. Such description includes the information of correlograms to learn at the same time the object local representation and the spatial relationship among its parts fragments.

6.2.1 Boosting landmarks

A common strategy to address the object detection problem is to model the object as an arrangement of its parts. The representative parts of the object (e.g. represented by a set of landmarks) must be highly discriminable; incorporating the spatial relationship between different parts [9] can improve significantly the robustness of the object detection.

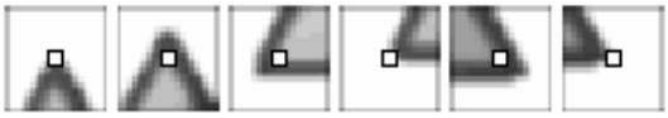
In order to avoid considering all possible ROIs of an image where an object can be located, first we define candidate locations of the object of interest by means of a set of landmarks. The set of object landmarks is selected manually from a data set. Using a training set of positive samples and a negative set of background image regions, we train each landmark using a cascade of classifiers [32]. In particular, Gentle Adaboost with Haar-like features estimated on the Integral Image [32] has been used in the cascade since it has been shown to outperform most of the other boosting variants in real applications [32]. Each level of the cascade is specialized on a complex set of features corresponding to a landmark. By adding cascade levels, the number of false positives is reduced while maintaining the detection of true positives, and the process is repeated for each landmark of the object. This approach has the advantage of reducing the number of landmark candidates when compared to other well-known techniques. For instance, Torralba et al. [90] use a set of masks and parts of an object and use normalized cross-correlation to obtain and detect the set of landmarks. By using the Haar-like features, compared to other methods like the normalized cross-correlation [90], we are more permissive to detect objects in case of object transformations and to obtain a lower level of confusion with the background regions. Summarizing, the steps to train a landmark detector are:

for each landmark:

- Define a positive set of image regions (centered in the landmark);
- Define a set of non-containing landmark images (negative set);
- Train a cascade of classifiers for each landmark.

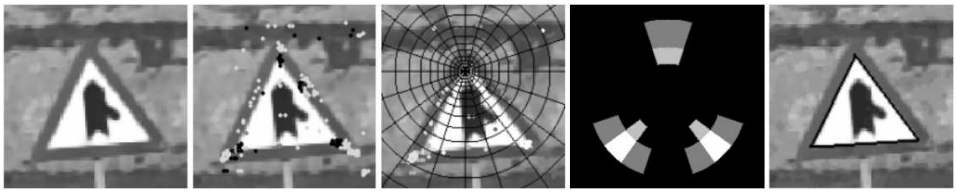
To illustrate the process observe the triangular traffic sign image in fig. 6.8(a). To distinguish this object type, we have manually identified six different object parts (landmarks) that can represent the object. The selected fragments are shown in fig.1. In the detection step, the set of selected landmarks is learnt using Gentle Adaboost with the Haar-like features estimated in the integral image. In particular,

for the example in fig. 6.8 we used 100 real triangular signs to generate a set of 100 positive samples of 21×21 pixels for each landmark. For each fragment, its 100 positives samples and 500 random background samples of the same size are trained in an attentional cascade of 10 levels, allowing a false alarm rate by stage of 30%. This measure assures that each landmark classifier has learnt correctly 100% of the positives samples, and the small number of detected false positives does not introduce ambiguity at the detection step. The use of six landmark cascades gives the results shown in fig. 6.8(b). We can observe that it has a small number of detected labeled landmarks compared to all possible locations and scales. Note that the presented scheme is quite robust to scale, translation, global illumination and to small object affine transformations, avoiding the problems of background confusion of masked landmarks because of the use of the Haar-like features.



(a) (b) (c) (d) (e) (f)

Figure 6.7: Selected landmarks for triangular signs.



(a) (b) (c) (d) (e)

Figure 6.8: (a) Input image. (b) Detected landmarks. (c) Contextual descriptors. (d) Resulting bins at feature selection of the correlogram of the landmark of fig. 6.7(b). (e) Detected sign.

6.2.2 Contextual Descriptors

In order to refine the set of landmarks we use their contextual description. This step focuses on defining the spatial relationship among the previously detected landmarks to be learnt. Our approach proposes an alternative point of view of the method of [7] in which a set of points of interest $P = \{p_i\}_{i=1}^{N}$ is considered, where N is the number of points of interest coming from the edges of the image. These points are used to build the constellation of multi-scale correlograms. However, opposed to the work presented in [7], our relevant information is provided by landmarks instead of a set of contour points. Since we are focusing on landmark candidates, we can exploit the previous knowledge about the relationship between the landmarks and the size of the

object of interest, reducing considerably the number of false positives and avoiding the multi-scale step. Considering n landmarks and their sets of detected candidates:

$$L^1 = \{L_1^1 ... L_{i_1}^1\}, ..., L^n = \{L_1^n ... L_{i_n}^n\} \tag{6.2}$$

where i_j is the number of instances of landmark j found in the image, for each combination of possible landmarks candidates:

$$\{L_{j_1}^1, ..., L_{j_n}^n, j_1 \in \{1, ..., i_1\}, ..., j_n \in \{1, ..., i_n\}\} \tag{6.3}$$

we generate n correlograms centered at the n chosen candidates. Their combination forms a constellation. From this constellation, we design a contextual descriptor vector:

$$D = (D_1, ..., D_n) \tag{6.4}$$

The descriptor vector associated to each landmark candidate is described by:

$$D_i = \{B_i^1, ..., B_i^n\} \tag{6.5}$$

being:

$$B_i^j = \{(o_j, h_j, x_j)\}_{i=1}^n \tag{6.6}$$

where o_j is the label identifying the part, h_j are the properties describing the part, and x_j is its spatial description in the image defined by its shape context [10], as shown in fig. 6.8(c). Hence, the spatial relationship vector is defined by the values of the correlogram bins for each of the landmarks. For example, using the 6 landmarks shown in fig. 6.7, the spatial descriptor vector for an object is $6 \times N$ bins in length, where N is the number of bins that forms each correlogram.

Given L correlograms of N bins, we create the object descriptor rearranging the bins as a vector of size $N \times L$. Since the constructed descriptor is, usually, very highly dimensional, we use Gentle Adaboost (as in the case of learning the landmarks) as a feature selection algorithm to reduce the dimensionality of the feature space and to learn the representative features of the object. In this way, the final classifier learns at the same time the features that correspond to relevant landmarks and their respective spatial relations. Note that we can also introduce extra information, such as the image contour map as an additional information to the boosted landmarks.

The detected landmarks involved in the detection step are shown in fig. 6.8(b). First, all candidates of each type of landmark (each positive detection of a given landmark) are sorted by their likelihood using the margin of the output of the Gentle Adaboost classifier. Afterwards, we select the first combination of landmarks (one of each type) that maximizes the sum of the likelihood. The individual vector descriptors of each set of selected landmarks are merged. In fig. 6.8(c), the correlogram applied to the landmark displayed in fig. 6.7(b) is shown. In fig. 6.8(d), the locations learnt for the correlogram of the same landmark are shown. The gray level of the bins of the correlogram corresponds to the importance assigned by the boosting procedure - which is intuitively related to the likelihood of the presence of the other landmarks in the descriptor of a current landmark. When a combined descriptor from a set of landmark

candidates is classified as positive using the trained Gentle Adaboost classifier, the object presence and the location of its landmarks are defined. In fig. 6.8(e) we can observe a detected object, which contextual descriptor, defined by the combination of detected landmarks, has been accepted as a positive example using the classifier based on boosted landmarks.

6.3 Object recognition discussion

In this chapter, we presented two novel strategies for object detection and description that can be useful in combination with the coding and decoding ECOC strategies presented in this tesis to deal with multi-class visual pattern recognition problems.

Concerning the suitability of the presented BSM scheme for object description, several benefits should be mentioned: The method is rotation invariant because of the use of the Hottelling transform and the area density adjustment. The method is also scaling and stretching invariant because of the use of the BSM grid. Moreover, the BSM descriptor is robust against symbols with rigid and elastic deformations since the size of the BSM grid defines the region of activity of the symbol shape.

The previous properties makes the BSM approach to be useful on those problems where the shape is a relevant feature to describe objects, and the use of the ECOC schemes allows the BSM descriptor to be extended to multi-class object recognition problems.

Moreover, there exists other applications where the Multi-class BSM scheme could also be applied. Many description techniques are applied on problems where a previous region detection is required. This type of applications use to detect circular regions to be described. In this case, the BSM descriptor should be applied to this type of problems since it provides a feasible way to robustly describe grey-level regions on real environments. In the same way, circular grids could also be defined to allow the BSM descriptor to be described on this type of applications. Another possible application consists in symbol spotting. Because of the good shape encoding and fast computation of the BSM descriptor, it can be applied to this type of applications. It only requires the descriptor to be included in a detection procedure, such as the one proposed by Viola & Jones [92] based on a cascade of detectors.

Concerning the Boosted Landmarks of Contextual Descriptors applicability, the strategy provides a fast and robust way to detect objects in clutter scenes. The method is able to learn simultaneously the most relevant object features and their relations, being potentially useful to deal with problems that suffer from small variations in scale, translation, global illumination, partial oclusions, and small affine transformations.

On next chapters, we show the suitability of the techniques presented in this chapter jointly with the multi-class ECOC designs presented at the previous sections to solve several real world visual pattern recognition problems.

Chapter 7

Applications

In this chapter, we present different real applications where our methodology is applied. First, we introduce and model two medical categorization problems: intravascular ultrasound tissue characterization and Chaga's disease categorization. Then, we present a Mobile Mapping System to detect and categorize a wide set of traffic signs in uncontrolled environments. Finally, different benchmarking data sets from public data and symbol recognition domains are learnt using the novel object detection and feature extraction methodology jointly with the new ECOC designs.

7.1 Intravascular Ultrasound Tissue Characterization

Cardiovascular diseases represented the first cause of sudden death in the occidental world [65]. Plaque rupture is one of the most frequent antecedent of coronary pathologies. Depending on the propensity to collapse, coronary plaque can be divided into stable and vulnerable plaque [13]. According to pathological studies, the main features of a stable plaque are characterized by the presence of a large lipid core with a thin fibrous cap. This last type of plaque can rupture generating thrombi followed by an intimal hyperplasia. Therefore, an accurate detection and quantification of plaque types represents an important subject in the diagnosis in order to study the nature and the plaque evolution to predict its final effect.

One of the most widely used diagnostic procedures consists of screening the coronary vessels employing Intravascular Ultrasound Imaging (IVUS). This technique yields a detailed cross-sectional image of the vessel allowing coronary arteries and their morphology to be extensively explored. This image modality has become one of the principal tools to detect coronary plaque. An IVUS study consists of introducing a catheter which shots a given number of ultrasound beams and collect their echoes to form an image. According with these echoes, three distinguishable plaques are considered in this type of images: calcified tissue (characterized by a very high echo-reflectivity and absorbtion of the ultrasound signal), fibrous plaque (medium echo-reflectivity and good transmission coefficient), and lipidic or soft plaque (characterized with very low reflectance of the ultrasound signal).

Despite the high importance of studying the whole coronary vessel, in clinical practice, this plaque characterization is performed manually in isolated images. Moreover, due to the variability among different observers, a precise manual characterization becomes very difficult to perform. Therefore, automatic analysis of IVUS images represents a feasible way to predict and quantify the plaque composition, avoiding the subjectivity of manual region classification and diminishing the characterization time in large sequences of images.

Given its clinical importance, automatic plaque classification in IVUS images has been considered in several research studies. The process can be divided into two stages: plaque characterization step which consists of extracting characteristic features in order to describe each tissue, and a classification step where a learning technique is used to train a classifier. In the first stage there are mainly two basic strategies: image-based approaches [99][74], and Radio Frequency (RF) signal analysis [41][58]. The main advantage of image-based methods is the availability of the images since they are the standard data source of the equipment. Additionally there is a high variety of descriptors which capture the spatial information of gray level values of a pixel together with its neighborhood in the image. On the other hand, characterization of RF signal has been proposed to take advantage of the raw IVUS signals. This data source avoids the introduction of artifacts from the pixel interpolation in the process of image formation. Due to the higher resolution of the unprocessed data, small regions of plaque could be distinguished.

In this section, we present an intravascular data set based on texture-based features, RF signals, combined features, and slope-based features to characterize the

different types of tissues.

7.1.1 Feature Extraction

We consider three types of features, the first ones obtained from RF signals, the second ones based on texture-based features from reconstructed images, and finally, the slope-based features proposed in [60].

RF Features

In order to analyze ultrasound images, the RF signals are acquired from the IVUS equipment with a sampling rate of at least two times the transducer frequency, and filtered using a band-pass filter with 50% gain centered at the transducer frequency [40]. Then, an exponential Time Gain Compensation (TGC) is applied [40]. Once the RF signals have been acquired, filtered and exponentially compensated by the TGC, the power spectrum is obtained. Nair et al. in [60] show the modelling of the power spectrum using Autoregressive Models (ARM) as one of the most suitable and stable methods to analyze ultrasound signals [60]. It also represents an alternative to the Fourier Transform since the ARM have been proved to be more stable when small signal windows are considered.

The ARM are defined as a linear prediction equation where the output x at a certain point t for each A-line is equal to a linear combination of its p previous outputs weighted by a set of parameters a_p [69]:

$$x(t) = \sum_{k=1}^{p} a_p(k)x(t-k),$$

where p is the ARM degree and the coefficients a_p are calculated minimizing the error of the modelled spectrum with respect to the original using the Akaike's error prediction criterium [69].

A sliding window is formed by n samples and m contiguous A-lines with a displacement of $n/4$ samples and $m/3$ A-lines in order to obtain an average AR model of a region. Only one side of the obtained spectrum is used because of its symmetrical properties. This spectrum is composed of h sampled frequencies ranging from 0 to $f_s/2$ [69].

In addition to the spectrum, two global measures are computed: the energy of the A-line and the energy of the window spectrum. All these features are compiled into a unique vector of $h+2$ dimensions which is used as a feature vector in the classification process.

Texture Features Extraction

Given that different plaques can be discriminated as regions with different grey-level distributions, it is a natural decision to use texture descriptors. In the bibliography, one can find a wide set of texture descriptors and up to our knowledge there are no optimal texture descriptors for image analysis in the general case. Our strategy is instead of trying to find out the optimal texture descriptor for our problem to gather

several families of descriptors and apply multiple classifiers able to learn and extract the optimal features for the concrete problem.

Therefore, we employ three different texture descriptors: co-occurrence Matrix [63], local binary patterns [64] and Gabor filters [20, 11]. Additionally, taking into account that highly non-echogenic plaques produce significant shade in the radial direction of the vessel, we include in the feature set the presence of shading in the image as a complementary feature.

The co-occurrence matrix is defined as the estimation of the joint probability density function of gray level pairs in an image [63]. The sum of all element values is:

$$P(i,j,D,\theta) = P(I(l,m) = i \otimes I(l + Dcos(\theta), m + Dsin(\theta)) = j),$$

where $I(l,m)$ is the gray value at pixel (l,m), D is the distance among pixels and θ is the angle between neighbors. We have established the orientation θ to be $[0^o, 45^o, 90^o, 135^o]$ [77, 63]. After computing this matrix, Energy, Entropy, Inverse Difference Moment, Shade, Inertia and Promenance measures are extracted [63].

Local Binary Patterns (LBP) are used to detect uniform texture patterns in circular neighborhoods with any quantization of angular space and spatial resolution [64]. LBP are based on a circular symmetric neighborhood of P members with radius R. To achieve gray level invariance, the central pixel g_c is subtracted to each neighbor g_p, assigning the value 1 to the result if the difference is positive and 0, otherwise. LBPs are defined as follows:

$$LBP_{R,P} = \sum_{p=0}^{P} a(g_p - g_c) \cdot 2^p$$

A Gabor filter is a special case of wavelets [20] which is essentially a Gaussian modulated by a complex sinusoid s. In 2D, it has the following form in the spatial domain:

$$\begin{array}{ll} h(x,y) = \frac{1}{2\pi\sigma^2} \exp\{-\frac{1}{2}[(\frac{x^2+y^2}{\sigma^2})]\} \cdot s(x,y) & \\ s(x,y) = \exp[-i2\pi(Ux + Vy)] & \phi = \arctan V/U \end{array}$$

where σ is the standard deviation, U and V represent the 2D frequency of the complex sinusoid, and ϕ is the angle of the frequency.

According to [34], one of the main differences in the appearance of calcified tissue compared to the rest of tissue types is the shadow which is appreciated behind it. In order to detect this shadow, we perform an accumulative mean of the pixels gray values on the polar image from a pixel to the end of the column (the maximal depth considered). As a result of extracting the texture descriptors, we construct an n-dimensional feature vector where $n = k + l + m + 1$, k is the number of co-occurrence matrix measurements, l is the number of Gabor filters, m is the number of LPB and the last feature is the measure of the "shadow" in the image.

Intravascular data set

In order to generate the data sets, we used the RF signals and their reconstructed images from a set of 10 different patients with Left Descent Artery pullbacks acquired in Hospital "German Trias i Pujol" from Barcelona, Spain. All these pullbacks contain

the three classes of plaque. For each one, 10 to 15 different vessel sections were selected to be analyzed. Two physicians independently segmented 50 areas of interest per pullback. From these segmentations we took 15 regions of interest (ROI) of tissue per study randomly making a total of 5000 evaluation ROIs. To build the data set, these selections were mapped in both RF signals and reconstructed images. In order to reduce the variability among different observers, the regions where both cardiologist agreed have been taken under consideration. Some samples from the data set are shown on the left of fig. 7.1.

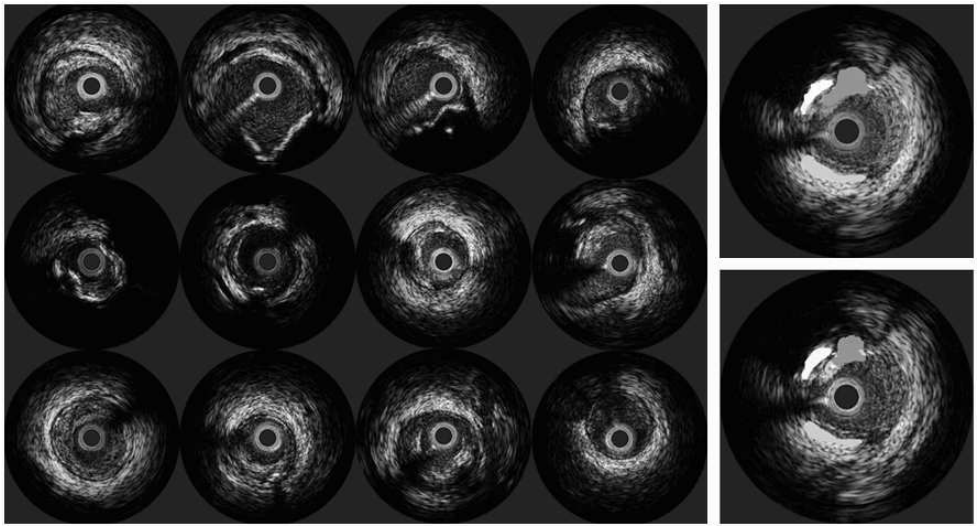


Figure 7.1: Left: IVUS data set samples. Right: (top) segmentation by a physician and (down) Automatic classification with Texture-Based Features. The white area corresponds to calcium, the light gray area to fibrosis, and the dark gray area to soft plaque.

To generate the data set on texture features, the intersection between segmented images is mapped into a feature vector. Then, all the features collected are categorized by patient and each of the three possible plaques type. The image features are extracted by using the previous texture descriptors: Co-ocurrence Matrix, Local Binary Patterns, and Gabor Filters. Those features are calculated for each pixel and gathered in a feature vector of 68 dimensions. An example of a manual and automatic texture-based segmentation for the same sample is shown on the right of fig. 7.1.

To generate the data set of RF features, the RF signals have been acquired using a 12-bit acquisition card with a sampling rate of $f_s = 200MHz$. The IVUS equipment used is Galaxy II from Boston Scientific with a catheter transducer frequency of $f = 40Mhz$, and it is assumed a sound speed in tissue of $1565m/s$. Each IVUS image consists of a total of 256 A-lines (ultrasound beams), with a radial distance of $r = 0.65cm$. The attenuation in tissue factor used is $\alpha = 1Db/Mhz \times cm$. To analyze the RF signals, the sliding window is composed of $n = 64$ samples of depth and $m = 12$ radial A-lines, and the displacement is fixed in 16 samples and four A-

lines. The power spectrum of the window ranges from 0 to $100MHz$ and it is sampled by 100 points. Then, it is complemented with two energy measures yielding a 102 feature vector.

We also consider a third data set that concatenates the descriptors from the previous RF and texture-based features, obtaining a feature vector of length 170 features.

Slope-based features

Finally, the fourth data set considers the slope-based features proposed by [60]. In particular, each sample is characterized by means of 14 slope-based features corresponding to: maximum power in DB from 20 to 60 MHz, frequency at the maximum power, negative slope in db/MHz between maximum and 60, minimum power in that slope, frequency corresponding to this negative slope, the estimated y intercept of this slope, the positive slope in db/Mhz between 20 and maximum, minimum power in that slope, frequency corresponding to this negative slope, the estimated y intercept of this slope, the mean power, the power at 0 MHz, power Db at 100 Mhz, and the power at the midband frequency (40 MHz) in DB [60].

7.1.2 Intravascular tissue characterization

To solve the problem of Intravascular tissue characterization, first we apply the Subclass ECOC strategy over the four previous data sets, and second, we apply an incremental tissue categorization using the decoding evaluation presented in this thesis.

IVUS characterization with sub-classes

For this experiment, we use the four previous IVUS data sets. To measure the performances, we apply leave-one-patient-out evaluation.

Applying *NMC*, Adaboost, and *FLDA* over a set of ECOC configurations, the performance results for RF features, texture-based features, combined RF and texture-based features, and slope-based features are shown in fig. 7.2. Comparing the results among the different data sets, one can see that the worst performances are obtained by the RF and slope-based features, which obtain very similar results for all the base classifiers and ECOC configurations. The texture-based features obtain in most cases results upon 90%. Finally, the data set of combined RF and texture-based features slightly outperform the results obtained by the texture-based feature, though the results do not significantly differ[1]. This behavior is summarize on table 7.1, where the mean rank obtained by each feature set is shown. The rankings are obtained estimating each particular ranking r_i^j for each problem i and each feature set j, and computing the mean ranking R for each feature set as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of problems (3 base classifiers $\times$ 6 ECOC designs). Note that the best ranking corresponds to the combined set of features, and that the individual feature set that obtains the best results correspond to texture-based.

Concerning the classification strategies, observing the obtained performances in fig. 7.2, one can see that independently of the data set and the ECOC design applied,

[1]Due to the high similitude among slope-based and RF features results, the combination of texture-based and slope-based features has been omitted.

Table 7.1: Mean rank for each feature set.

Feature set	RF	Texture-based	RF+Texture-based	Slopes
Mean rank	2.94	2.28	1.72	2.83

the Sub-class ECOC approach always attains the best results. To compare these performances, the mean rank of each ECOC design considering the twelve different experiments is shown in table 7.2. In this case, the rankings are obtained estimating each particular ranking r_i^j for each problem i and each ECOC configuration j, and computing the mean ranking R for each ECOC design as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of problems (3 base classifiers $\times$ 4 data sets). One can see that the Sub-class ECOC attains the best position for all experiments. To analyze if the difference between methods ranks are statistically significant, we apply the Friedman and Nemenyi tests. In order to reject the null hypothesis that the measured ranks differ from the mean rank, and that the ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}] \tag{7.1}$$

In our case, with $k = 6$ ECOC designs to compare, $X_F^2 = 30.71$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{7.2}$$

Applying this correction we obtain $F_F = 11.53$. With six methods and twelve experiments, F_F is distributed according to the F distribution with 5 and 55 degrees of freedom. The critical value of $F(5, 55)$ for 0.05 is 2.40. As the value of F_F is higher than 2.45 we can reject the null hypothesis. One we have checked for the for the non-randomness of the results, we can perform a post hoc test to check if one of the techniques can be singled out. For this purpose we use the Nemenyi test - two techniques are significantly different if the corresponding average ranks differ by at least the critical difference value (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{7.3}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$. In our case, when comparing six methods with a confidence value $\alpha = 0.10$, $q_{0.10} = 1.44$. Substituting in eq.7.3, we obtain a critical difference value of 1.09. Since the difference of any technique rank with the Sub-class rank is higher than the CD, we can infer that the Sub-class approach is significantly better than the rest with a confidence of 90% in the present experiments.

Table 7.2: Mean rank for each ECOC design over all the experiments.

ECOC design	one-versus-one	one-versus-all	dense random
Mean rank	2.33	5.08	4.25
ECOC design	sparse random	decoc	sub-class
Mean rank	5.00	2.67	1.00

IVUS characterization with decoding evaluation

Given the high variability of IVUS problem, we consider a multi-patient classification strategy for this experiment. Starting from three arbitrary plaques that can belong to different patients, and increasing the set of plaques by one, the one-versus-one strategy is applied to see the performance of each decoding strategy in this problem. All the decoding strategies presented in this thesis are considered in this experiment with Gentle Adaboost as base classifier. Once a classification is done among all patient plaques, label by its corresponding tissue. The classification results are shown in fig. 7.3. In this experiment, one can see that the classification performance tends to increase when the number of classes also increases. The results of most of the decoding strategies are highly correlated. Only the ELW and LLW strategies obtain different results from the rest of approaches. The continuous ELW attains the lowest accuracy on this problem in comparison with the others. Finally, the continuous LLW is the most appropriated choice in this case, outperforming at each step of the experiment (thus, for any number of plaques) the results obtained by the rest of strategies.

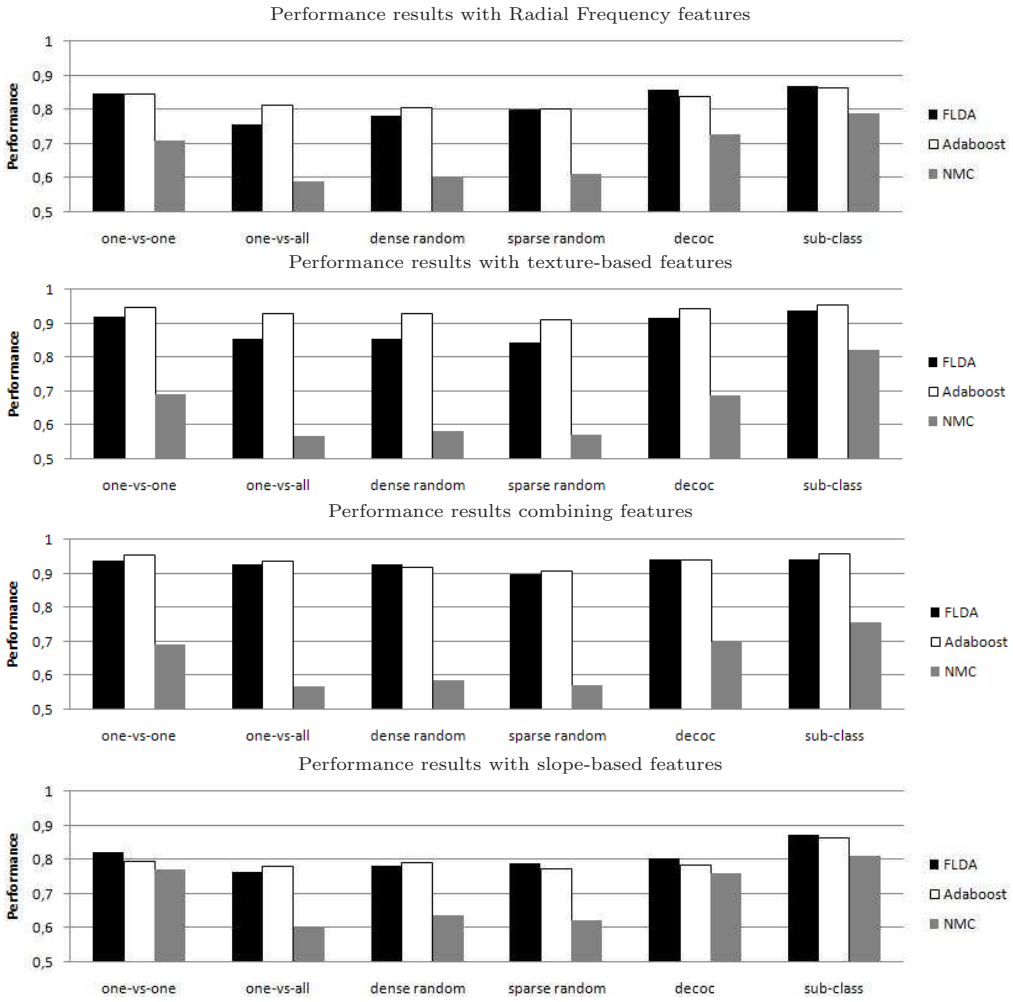


Figure 7.2: Performance results for different sets of features, ECOC designs and base classifiers on the IVUS data set.

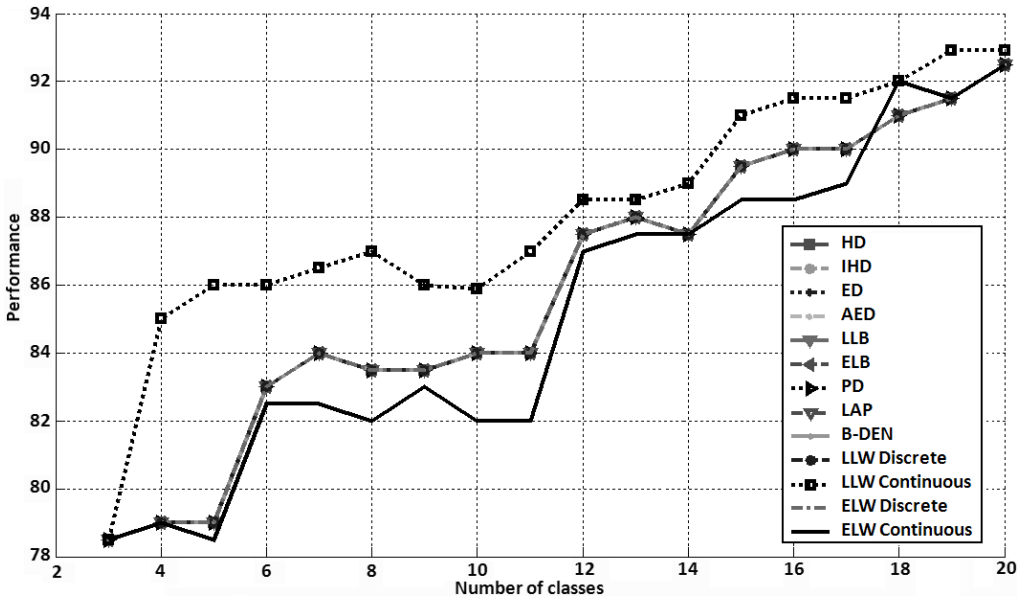


Figure 7.3: Classification results for the decoding strategies when the size of the training data increases.

7.2 Chagas' disease

Chagas' disease is an infectious illness caused by the parasite Tripanosoma Cruzi, which is transmitted to humans through the feces of a bug called Triatoma infestans. The most common insect species belong to the genera Triatoma (fig. 7.2(a)), Rhodnius (fig. 7.2(b)), and Panstrongylus.

(a) (b)

Figure 7.4: (a) Triatoma and (b) adult Rhodnius prolixus, a kissing bug.

Trypanosoma cruzi is a member of the same genus as the infectious agent of the African sleeping sickness and the same order as the infectious agent of leishmaniases, but its clinical manifestations, geographical distribution, life cycle and insect vectors are quite different. The Chagas disease is endemic in all Latin America, and according to the studies of the OMS, about a 18 millions of people suffer from this disease in the continent, 100 million (25% of the Latin American population) have risk of acquiring the disease, and around 50.000 infected people annually die [62]. The black regions of fig. 7.5 correspond to the geographical Latin American areas affected by the Chagas disease.

Figure 7.5: Geographic influence of the Chagas disease in Latin American.

An infected triatomine insect vector feeds on blood and releases trypomastigotes in its feces near the site of the bite wound. The victim, by scratching the site of the bite, causes trypomastigotes to enter the host through the wound, or though intact mucosal membranes, such as the conjunctiva. Then, inside the host, the trypomastigotes invade cells, where they differentiate into intracellular amastigotes. The amastigotes multiply by binary fission and differentiate into trypomastigotes, then are released

into the circulation as bloodstream trypomastigotes (fig. 7.6). These trypomastigotes infect cells from a variety of biological tissues and transform into intracellular amastigotes in new infection sites.

Figure 7.6: Tripomastigote and bloodstream trypomastigotes.

In general terms, two different stages of Chagas' disease can be distinguished. The first stage, called acute phase, appears shortly after the parasitical infection and it is occasionally manifested by high temperature, inflammations, and heart rate acceleration. Following this phase, which lasts for one or two months, there is an undetermined latent period. After that, some patients go into a chronic phase, which is characterized by alterations in the cardiovascular system, normally associated to the so-called Chagas' cardiomyopathy. This type of cardiomyopathy produces malfunctioning in the propagation of the electrical impulse as well as destruction of cardiac fibers. In areas where the illness is endemic, Chagas' cardiomyopathy represents the first cause of cardiovascular death [51].

In order to optimize treatment for chronic chagasic patients, it is essential to make use of an effective diagnosis tool able to determine the existence of cardiac injury and, if positive, its magnitude. Clinical diagnosis is usually based on tests such as chest x-rays, echocardiogram, or electrocardiogram (ECG), which can be either Holter ECG or conventional rest ECG. The use of high-resolution electrocardiography (HRECG) has been reported in the literature as a useful tool for clinical assessment of Chagas' disease [14][26][55]. Specifically, the presence of ventricular late potentials (VLP) has been detected in chronic chagasic patients using high-resolution ECGs. VLP, which are usually measured on temporally averaged beats, are very low-amplitude high-frequency signals found within the terminal part of the QRS complex and the beginning of the ST segment. A different approach has been proposed in other studies [45][46], in which the beat-to-beat variability of the QRS duration on HRECG has been measured, and it has been shown that such a variability is more accentuated in chagasic patients, particularly when the degree of myocardial damage is severe.

Since Chagas' cardiomyopathy frequently leads to alterations in the heart's electrical conduction, recently it has been proposed the slopes of QRS complex in order to determine the myocardial damage associated with the disease [71].

QRS features

To obtain the features to evaluate the degree of myocardial damage associated with

the disease, the QRS slopes are analyzed for all the HRECG recordings of 107 individuals from the Chagas data set recorded at Simón Bolívar University (Venezuela). For each recording, let's denote $x_i(n), n = 0, ..., N$, the i-th beat of lead X, where i runs from 0 to I (being I the total number of beats in the recording). Analogously, let's denote $y_i(n)$ and $x_i(n)$ the i-th beats of leads Y and Z, respectively. QRS slopes are measured on temporally averaged signals $\overline{x}(n)$, $\overline{y}(n)$, and $\overline{z}(n)$, $n = 0, .., N$, which are calculated as the average of all normal beats $i = 0, .., I$ of the recording. Ectopic and grossly noisy beats were excluded of the averaging process. The averaging is performed following the standard recommendations described in [12].

A three-step process is applied to compute the upward QRS slope, α_{US}, and the downward QRS slope, α_{DS}, of each averaged beat $\overline{x}(n)$, $\overline{y}(n)$, and $\overline{z}(n)$. In the first step, delineation is performed using a wavelet-based technique [53] that determines the temporal locations Q, R, and S wave peaks, which are denoted by n_Q, n_R, and n_S, respectively [72]. The second step identifies the time instant n_U associated with maximum slope of the ECG signal (i.e., global maximum of its derivative) between n_Q and n_R. Analogously, the time instant n_D corresponding to minimum slope of the ECG signal between n_R and n_S is identified. As a final step, a line is fitted in the least squares sense to the ECG signal in a window of 15ms around n_U, and the slope of that line is defined as α_{US}. In the same manner, α_{DS} is defined as the slope of a line fitted in a 15ms window around n_D.

Other temporal indices defined to detect the presence of VLP in HRECG recordings are also evaluated in this work. Previous studies in the literature have shown the ability of those indices to determine the severity of Chagas' cardiomyopathy [45][46]. Consequently, we use such indices in conjunction with the QRS slopes. Computation of QRS-based indices considers filtered leads X, Y, and Z using a bi-directional 4th-order Butterworth filter with passband between 40 and 250 Hz. The filtered signals are denoted by $x_{i,f}(n)$, $y_{i,f}(n)$, and $z_{i,f}(n)$.

The QRS-based indices $QRSD$, $RMS40$, and $LAS40$, which are described next, require temporal signal averaging ($\overline{x_f}(n)$, $\overline{y_f}(n)$, and $\overline{z_f}(n)$) as well as the calculation of the vector magnitude, defined as follows:

$$v(n) = \sqrt{\overline{x_f}^2(n) + \overline{y_f}^2(n) + \overline{z_f}^2(n)} \tag{7.4}$$

On the signal $v(n)$ the three temporal QRS indices defined to detect VLP are computed based on identification of time instants n_b and n_e corresponding to the beginning and the end of the QRS complex [12]:

$$QRSD = n_e - n_b \tag{7.5}$$

$$RMS40 = \sqrt{\frac{1}{n_2 - n_1} \sum_{n=n_1}^{n_2} v^2(n)}, n_1 = n_e - 40ms, n_2 = n_e \tag{7.6}$$

$$LAS40 = n_e - argmax\{n | v(n) \geq 40\mu V\} \tag{7.7}$$

On the other hand, the index $\Delta QRSD$ is considered, which is defined next. This index is measured on the vector magnitude of the ultraveraged filtered leads ($x_{i,f}(n)$,

$y_{i,f}(n)$, $z_{i,f}(n)$):

$$v_i(n) = \sqrt{x_{i,f}^2(n) + y_{i,f}^2(n) + z_{i,f}^2(n)} \tag{7.8}$$

On each signal $v_i(n)$, $i = 0, .., I$, the duration of its complex QRS is estimated and denoted by $QRSD_i$. The index $\Delta QRSD$ is defined as the standard deviation of the beat-to-beat $QRSD_i$ series [46]:

$$\Delta QRSD = \sqrt{\frac{\sum_{i=1}^{I}(QRSD_i - \overline{QRSD}^2)}{I-1}}, \overline{QRSD} = \frac{\sum_{i=1}^{I} QRSD_i}{I} \tag{7.9}$$

Data set

We analyzed a population composed of 107 individuals from the Chagas data set recorded at Simón Bolívar University (Venezuela). For each individual, a continuous 10-minute HRECG was recorded using orthogonal XYZ lead configuration. All the recordings were digitalized with a sampling frequency of 1 kHz and amplitude resolution of 16 bits.

Out of the total 107 individuals of the study population, 96 are chagasic patients with positive serology for Trypanosoma Crucy, clinically classified into three different groups according on their degree of cardiac damage (Groups I, II, and III). This grouping is based on the clinical history, Machado-Guerreiro test, conventional ECG of twelve derivations, Holter ECG of 24 hours, and myocardiograph study for each patient. The other 11 individuals are healthy subjects with negative serology taken as a control group (Group 0). All individuals of the data set are described with a features vector of 16 features based on the previous analysis of section 2. The four analyzed groups are described in detail next:

• Group 0: 11 healthy subjects in the age 33.6±10.9 years, 9 men and 2 women.

• Group I: 41 total patients with the Chagas' disease in the age of 41.4±8.1 years, 21 men and 20 women, but without evidences of cardiac damage in cardiographic study.

• Group II: 39 total patients with the Chagas' disease in the age of 45.8±8.8 years, 19 men and 20 women, with normal cardiographic study and some evidences of weak or moderate cardiac damage registered in the conventional ECG or in the Holter ECG of 24 hours.

• Group III: 16 total patients with the Chagas' disease in the age of 53.6±9.3 years, 9 men and 7 women, with significant evidences of cardiac damage detected in the conventional ECG, premature ventricular contractions and/or cases of ventricular tachycardiac registered in the Holter ECG and reduced fraction of ejection estimated in the cardiographic study.

7.2.1 Chagas' disease characterization

We compare our results with the performances reported in [71] for the previous data. Moreover, we compare different ECOC designs: the one-versus-one ECOC coding strategy applied with the Hamming, Euclidean, Probabilistic, and the presented Loss-Weighted decoding strategies. We selected the one-versus-one ECOC coding strategy because the individual classifiers are usually smaller in size than they would be in the rest of ECOC approaches, and the problems to be learned are usually easier, since the

classes have less overlap. Each ECOC configuration is evaluated for three different base classifiers: Fisher Linear Discriminant Analysis (*FLDA*) with a previous 99.9% of Principal Components, Discrete Adaboost with 50 runs of Decision Stumps, and Linear Support Vector Machines with the regularization parameter C set to 1. To evaluate the methodology we apply leave-one-patient-out classification on the Chagas data set.

We divide the Chagas categorization problem into two experiments. First, we classify the features obtained from the 107 patients considering the four groups in a leave-one-patient-out experiment for the different ECOC configurations and base classifiers. Since each patient is described with a vector of 16 features, 107 tests are performed. And second, the same experiment is evaluated over the 96 patients with the Chagas' disease from groups I, II, and III. This second experiment is more useful in practice since the splitting of healthy people from the patients with the Chagas' disease is solved with an accuracy upon 99.8% using the Machado-Guerreiro test.

4-class characterization

The results of categorization for the four groups of patients reported by [71] are shown in fig. 7.7. Considering the number of patients from each group, the mean classification accuracy of [71] is of 57%. The results using the different ECOC configurations for the same four groups are shown in fig. 7.8. In fig. 7.8(a), the mean accuracy for each base classifier and decoding strategy is shown. The individual performances of each group of patients for each base classifier are shown in fig. 7.8(b), fig. 7.8(c), and fig. 7.8(d), respectively. Observing the mean results of fig. 7.8(a), one can see that any ECOC configuration outperforms the results reported by [71]. Moreover, even if we use *FLDA*, Discrete Adaboost, or Linear *SVM* in the one-versus-one ECOC design, the best performance is always obtained with the proposed Loss-Weighted decoding strategy. In particular, the one-versus-one ECOC coding with Discrete Adaboost as the base classifier and Loss-Weighted decoding attains the best performance, with a classification accuracy upon 60% considering the four groups of patients.

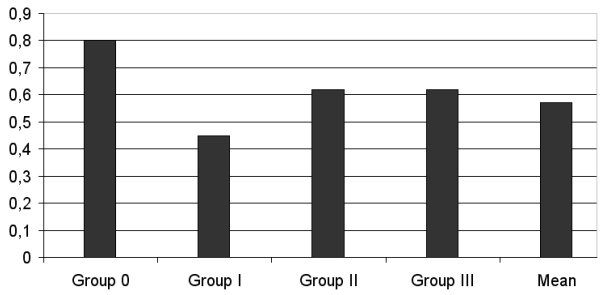
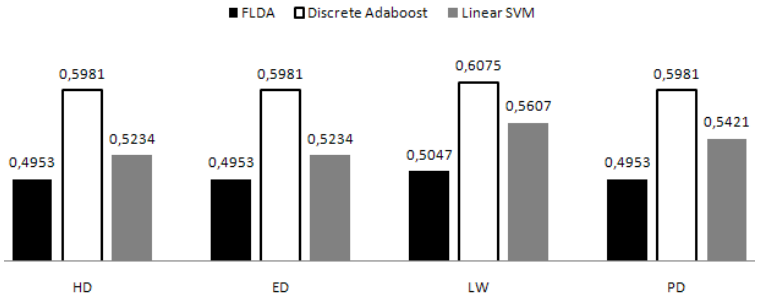


Figure 7.7: Classification performance reported by [71] for the four groups of patients.

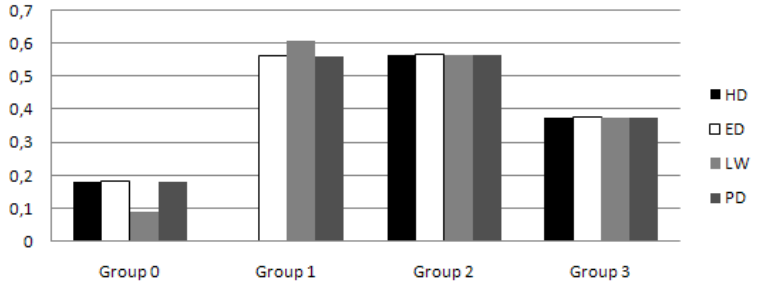
3-class characterization

Now, we evaluate the same strategies on the three groups of patients with the Chagas' disease, without considering the healthy people. The new results are shown in fig. 7.9. In fig. 7.9(a), the mean accuracy for each base classifier and decoding strategy is shown. The individual performances of each group of patients for each base classifier are shown in fig. 7.9(b), fig. 7.9(c), and fig. 7.9(d), respectively. In the mean results of fig. 7.9(a), one can see that independently of the base classifier applied, the Loss-Weighted decoding strategy attains the best performances. In this example, the one-versus-one ECOC coding with Discrete Adaboost as the base classifier and Loss-Weighted decoding also attains the best results, with a classification accuracy about 72% distinguishing among three levels of patients with the Chagas' disease.

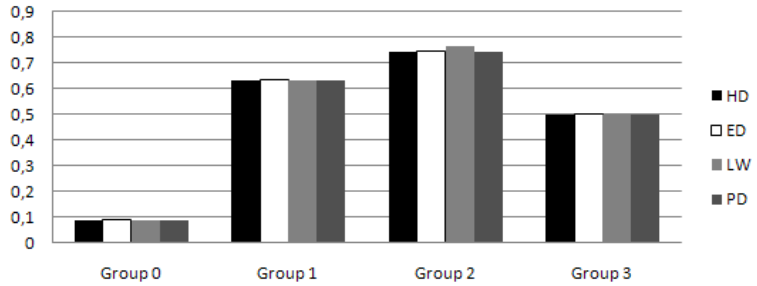
(a) Mean classification performance for each base classifier



(b) Classification performance for each group using *FLDA*



(c) Classification performance for each group using Discrete Adaboost



(d) Classification performance for each group using Linear *SVM*

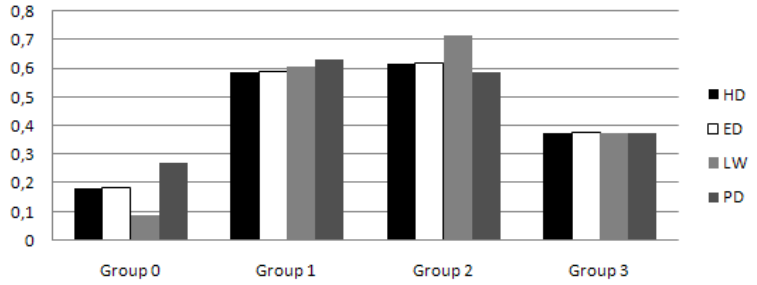
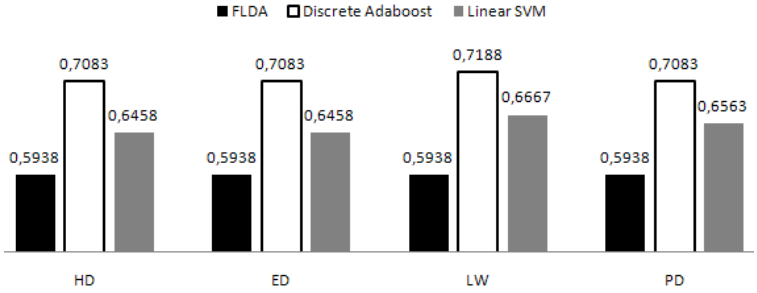
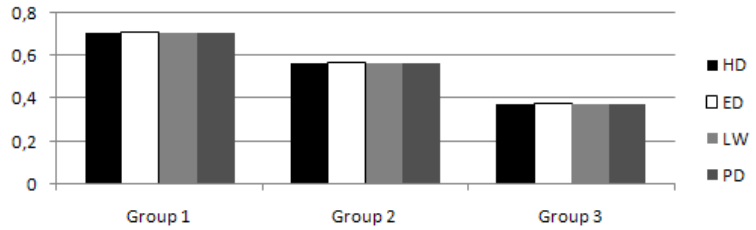


Figure 7.8: Leave-one-patient-out classification using one-versus-one ECOC design (HD: Hamming decoding, ED: Euclidean decoding, LW: Loss-Weighted decoding, PD: Probabilistic decoding) for the four groups with and without Chagas' disease.

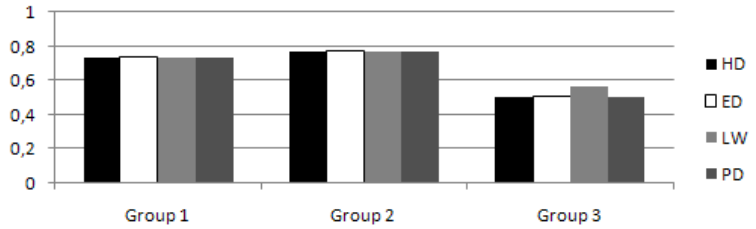
(a) Mean classification performance for each base classifier



(b) Classification performance for each group using *FLDA*



(c) Classification performance for each group using Discrete Adaboost



(d) Classification performance for each group using Linear *SVM*

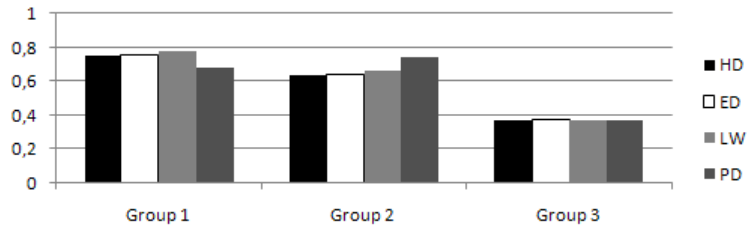


Figure 7.9: Leave-one-patient-out classification using one-versus-one ECOC design (HD: Hamming decoding, ED: Euclidean decoding, LW: Loss-Weighted decoding, PD: Probabilistic decoding) for the three groups with Chagas' disease.

7.3 Mobile Mapping System

We use the video sequences obtained from the Mobile Mapping System [16] to design a real traffic sign multi-class data set.

7.3.1 Data acquisition

In this system, the position and orientation of the different traffic signs are measured with video cameras fixed on a moving vehicle (see fig. 7.10). The system has a stereo pair of calibrated cameras, which are synchronized with a GPS/INS system. The result of the acquisition step is a set of stereo-pairs of images with their position and orientation information. We focus on the speed data set since the low resolution of the images, the non-controlled conditions, and the high similarity among classes make the categorization of these signs a difficult task.

Figure 7.10: Geovan.

We use Adaboost [32] to train a cascade that defines regions of interest (ROI) containing a sign. Depending on the type of the detected sign, a different model fitting is applied, looking for affine transformations that perform the spatial normalization of the object.

7.3.2 Model fitting

Because of the few changes on the point of view of the captured signs, we apply the fast radial symmetry [50] for the circular signs, which offers high robustness against image noise. As it is shown if Figure 7.11, the fast radial symmetry provides an approximation to the center and the radius of the circular sign.

Figure 7.11: (a) Input image, (b) X-derivative, (c) Y-derivative, (d) image gradient, (e) accumulator of orientations, (f) center and radius of the sign.

On the other hand, for the case of triangular signs, the method that allows a successful model fitting is based on the Hough transform [56]. Nevertheless, we need to consider additional constraints to obtain the three representative border lines of a triangular traffic sign. Each line has associated a position in relation to the others. In Figure 7.12(a) a false horizontal line is shown. Since this line does not fulfil the expected spatial constraints of the object, we iterate the Hough procedure to detect the next representative line in the allowed range of degrees. The corrected image is shown in Figure 7.12(b). Once we have the three detected lines, we calculate their intersection, as shown in Figure 7.12(c). To assure that the lines are the expected ones, we complement the procedure looking for a corner at the circular region of each intersection surroundings (as shown in Figure 7.12(d) and (e)) $S = \{(x_i, y_i) \mid \exists p < ((x - x_i)^2 + (y - y_i)^2 - r^2)\} \mid i \in [1, ..., 3]$, where S is the set of valid intersection points, and p corresponds to a corner point to be located in a neighborhood of the intersection point.

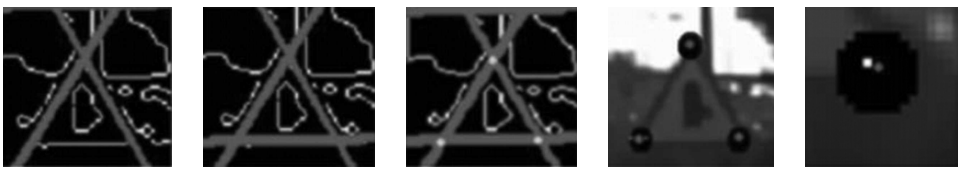


Figure 7.12: (a) Detected lines, (b) corrected line, (c) intersections, (d) corner region, (e) corner found.

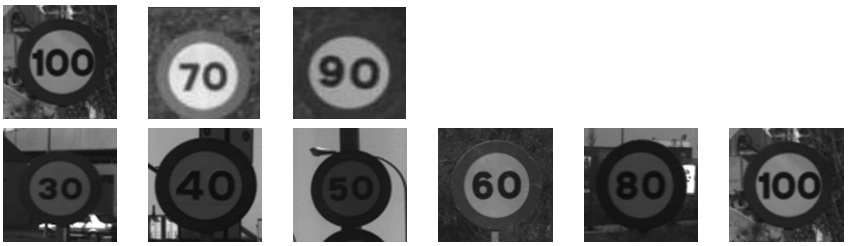
7.3.3 Spatial normalization

Once the sign model is fitted using the previous methods, the next step is the spatial normalization. The steps are: a) transform the image to make the recognition invariant to small affine deformations, b) resize the object to the signs data set size, c) filter using the Weickert anisotropic filter [93], and d) mask the image to exclude the background pixels at the classification step. To prevent the effects of illumination changes, the histogram equalization improves image contrast and yields an uniform histogram.

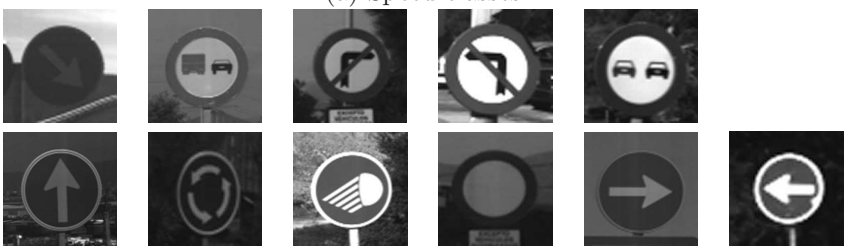
Figure 7.13: Samples from the road video sequences.

7.3.4 Traffic signs data set

Figure 7.13 shows examples of video sequences. We defined three groups of classes using the most common types of signs. The considered classes are shown in Figure 7.14. Speed signs need special attention. These types of signs are less discriminative, being some of them only differentiated by a few pixels. With this type of signs it is better to work on binary images to avoid the errors that can be accumulated because of the grey levels of the signs. For the twelve classes of circular signs and twelve of triangular signs we have 750 training images in both cases. For the nine speed classes we use 600 training samples. Finally, the resolution of each data set is: 35×35 pixels for the circular group, 44×39 pixels for the triangular group, and 41×41 pixels for the speed group, respectively.



(a) Speed classes



(b) Circular classes

(c) Triangular classes

Figure 7.14: Set of classes considered in the classification module.

7.3.5 Mobile Mapping System characterization

First, we evaluate the Forest-ECOC methodology with the state-of-the-art classifiers to solve the traffic sign categorization problem. Moreover, an analysis of the tree structure embedding in the Forest-ECOC matrix for the Mobile Mapping System is shown. Then, we classify the Speed data set applying the Sub-class ECOC strategy. The Speed categorization is also used to evaluate the decoding methodology presented in this thesis. Finally, the same data set is used to evaluated the Sparse random designs presented in this thesis.

Forest-ECOC and state-of-the-art comparison

To evaluate the Forest-ECOC performance, we compare it with the state-of-the-art classifiers. The details for each strategy are: 3-Euclidean distance Nearest neighbors (K-NN), Tangent Distance (TD) [86] with invariant tangent vector with respect to translation, rotation, and scaling, 99.98% of Principal Components Analysis followed by 3-Nearest neighbors (PCA K-NN) [3], Fisher Linear Discriminant Analysis with

a previous 99.98% PCA (FLDA) [3], Support Vector Machine with projection kernel Radial Basis Function and the parameter $\gamma = 1$ (SVM) [37], Gentle Adaboost with decision stumps using the Haar-like features (BR) [32], multi-class Joint Boosting with decision stumps (JB) [90], Gentle Adaboost [32] Sampling with FLDA (BS), statistical Gentle Naive Boosting with decision stumps (NB) [32], and our Forest-ECOC (F-ECOC) with 3-embedded optimal trees using a 99% FLDA as a base classifier. In the different variants of boosting we apply 50 iterations. We use Gentle Adaboost since it shown to outperform the other Adaboost variants in real applications [32]. Table 7.3 shows the characteristics of the previous data used for the classification experiments.

Table 7.3: Characteristics of the data sets used for classification.

Dataset	#Training examples	#Test examples	#Features	#Classes
Circular	750	200	1225	12
Speed	500	200	1681	7
Triangular	750	200	1716	12

The classification results are shown graphically in Figure 7.15 for the different groups. One can see that the Forest-ECOC using FLDA as a base classifier attains the highest accuracy in all cases. Nevertheless, for the circular and triangular signs the differences among classifiers are significatively different because of the high discriminability of these two groups. The speed group is a more difficult classification problem. The numerical results for this group are shown in table 7.4. In this case, the Forest-ECOC strategy obtains an accuracy upon 90%, outperforming the rest of classifiers.

Table 7.4: Classification results for the Speed group.

Classification technique	Accuracy
K-NN	70.53±1.70
TD	46.37±2.30
PCA K-NN	68.23±1.50
FLDA	88.89±1.30
SVM	79.84±2.00
BR	85.93±2.10
JB	80.36±1.50
BS	88.85±1.90
NB	82.78±1.80
Forest-ECOC	**91.73±1.10**

Tree embedding analysis

The training evolution of the Forest-ECOC at the previous experiment is shown in Figure 7.16 for the speed group. Each iteration of the figure shows the classification accuracy by embedding a new node (binary classifier) from each optimal tree in the Forest-ECOC matrix M. The three optimal trees are split by the dark vertical lines. The respective trees are shown in Figure 7.17. In the first generated tree of Figure 7.17, one can see that the most difficult partitions are reserved to the final

classifiers of the tree. The next trees select the following best partitions of classifiers to avoid repeating classifiers. These classifiers learn sub-groups of classes from the same data, improving the classification results (Figure 7.16) by sharing their knowledge among classes.

Speed signs categorization with sub-classes

For this experiment, we choose the previous speed data set to evaluate the Sub-class strategy since the low resolution of the image, the non-controlled conditions, and the high similarity among classes make the categorization a difficult task. Fig. 7.18 shows several samples of the speed data set used for the experiments. The data set contains a total of 2500 samples divided into nine classes. From the original feature space, about 150 features are derived using a *PCA* that retained 90% of the total variance.

The performance and the estimated ranks using the different ECOC strategies for the different base classifiers are shown in table 7.5. These results are also illustrated in the graphics of fig. 7.19. One can see that in this particular problem, the sub-class is only required for Discrete Adaboost and NMC, while the rest of base classifiers are able to find a solution for the training set without the need for sub-classes. In this case, *RBF SVM* obtains low performances, and parameter optimization should be applied to improve these results. Nevertheless, it is out of the scope of this paper. Finally, though the results do not significantly differ between the strategies, the Sub-class ECOC approach attains a better position in the global rank of table 7.5.

Table 7.5: Rank positions of the classification strategies for the Speed data set.

	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
D. Adaboost	66.1(3.1)	56.6(3.1)	55.2(2.8)	52.3(3.6)	58.6(3.2)	60.8(3.1)
NMC	60.7(3.2)	50.65(3.7)	47.4(3.8)	45.1(3.8)	51.9(3.2)	62.8(3.1)
FLDA	74.7(2.8)	71.4(2.9)	74.9(2.6)	72.7(2.5)	72.6(2.8)	76.2(3.0)
Linear SVM	74.9(2.7)	72.3(2.1)	71.8(2.1)	68.2(2.9)	78.9(2.1)	78.9(1.9)
RBF SVM	45.0(0.9)	45.0(0.9)	45.0(0.9)	44.0(0.9)	45.0(0.9)	45.0(0.9)
Global rank	1.8	3.6	3.4	4.6	2.6	1.2

Speed signs categorization with decoding evaluation

For this experiment, we also choose the previous speed data set to evaluate the decoding methodology. For this experiment, we apply ten-fold cross-validation over the set of coding and decoding designs.

The rankings obtained from the experiments are shown in fig. 7.20. The performances from which the rankings are computed are shown in tables E.1 and E.2 of Appendix E. Note that the different variants of Loss-Weighted strategy obtain the best positions in this real experiment. In particular, the Exponential Loss-Weighted decoding using the continuous output of the base classifiers attains the best positions either when we use Gentle Adaboost as well as Linear SVM. The rest of Type III strategies obtain good performance too. This behavior is more significant if we observe the rankings without considering the confidence interval of fig. 7.20, corresponding to the second and fourth column of each group.

Speed signs categorization with Sparse designs

For this experiment, we use the same Speed data set used in the previous experiments. For this experiment, we applied the same random criteria than at the Sparse random chapter, with a length of codewords of nine, as the number of classes.

Table 7.6 shows the performance results on the Speed traffic data set for the Sparse Random designs using Gentle Adaboost and Linear SVM, respectively. The results on the top correspond to the performance and confidence interval using the classical Sparse Random strategy. The results on the bottom correspond to the results using the Sparse Random selection based on maximizing the new ternary distance. Note that in all cases, the results obtained by the new Sparse designs outperform the performances obtained by the classical approach.

Table 7.6: Classical Sparse Random results (performances on the top of each data set) and Sparse Random with ternary distance maximization (performances on the bottom of each data set) using Adaboost and SVM on the Speed traffic sign data set.

	HD	IHD	ED	AED	LLB	ELB
Adaboost	0.526	0.483	0.516	0.514	0.404	0.430
	0.041	0.043	0.047	0.044	0.031	0.029
	0.539	0.508	0.557	0.537	0.533	0.553
	0.030	0.034	0.029	0.028	0.037	0.032
SVM	0.629	0.531	0.605	0.633	0.656	0.662
	0.048	0.048	0.054	0.049	0.053	0.058
	0.668	0.619	0.646	0.675	0.656	0.697
	0.035	0.041	0.036	0.032	0.045	0.029
PD	LAP	βDEN	LLW Discrete	LLW Continuous	ELW Discrete	ELW Continuous
0.561	0.524	0.526	0.528	0.450	0.539	0.492
0.055	0.047	0.047	0.044	0.035	0.041	0.039
0.570	0.547	0.547	0.546	0.551	0.548	0.564
0.031	0.027	0.027	0.033	0.041	0.030	0.038
0.650	0.640	0.640	0.648	0.661	0.642	0.678
0.043	0.055	0.056	0.055	0.045	0.056	0.057
0.659	0.706	0.706	0.706	0.669	0.706	0.711
0.031	0.036	0.036	0.036	0.035	0.035	0.029

To show the performance improvements by selecting the new Sparse Random matrix, the absolute and relative improvements are shown in fig. 7.21 for Gentle Adaboost and Linear SVM, respectively. The light bars correspond to the relative improvement, and the dark lines to the absolute one. In this experiment, one can see that the ternary sparse maximization criterion also obtains performance improvements for all decoding strategies.

Tringular sign detection using Boosted Landmarks with Contextual Descriptors

Some samples of the triangular sign data set illustrating the variation of appearance of the signs are shown in fig. 7.22. Observe the high variability of the signs due to the non-controlled conditions of acquisition. In this experiment, triangular signs are detected using the boosted landmarks technique. The landmarks are located at the three corners of the signs. In order to learn each landmark at size 21×21 pixels (see previous fig. 6.7) we use cascades of classifiers. We used stratified ten-fold cross-validation to train each cascade of 10 levels and 500 negatives samples, with an expected error of 0.3. The correlograms used have a diameter of 150 pixels, 20 radius regions and 13 geometric circles with an enlarging factor of 1.3. As a result, we obtain a total of 780 features for each landmark correlogram including the object attributes and spatial positions.

Finally, we use the images from the described Mobile Mapping System to detect triangular traffic signs. The whole process of detection and recognition is illustrated for some test images in fig. 7.23. First column corresponds to the detected landmark candidates labeled by color. The second column of the figure shows the combination of landmarks obtained by the highest likelihood of the contextual descriptors classifiers. At the third column one can see the final recognition results - the recovered object from the traffic sign data set using the Forest-ECOC strategy described in section 3.1.

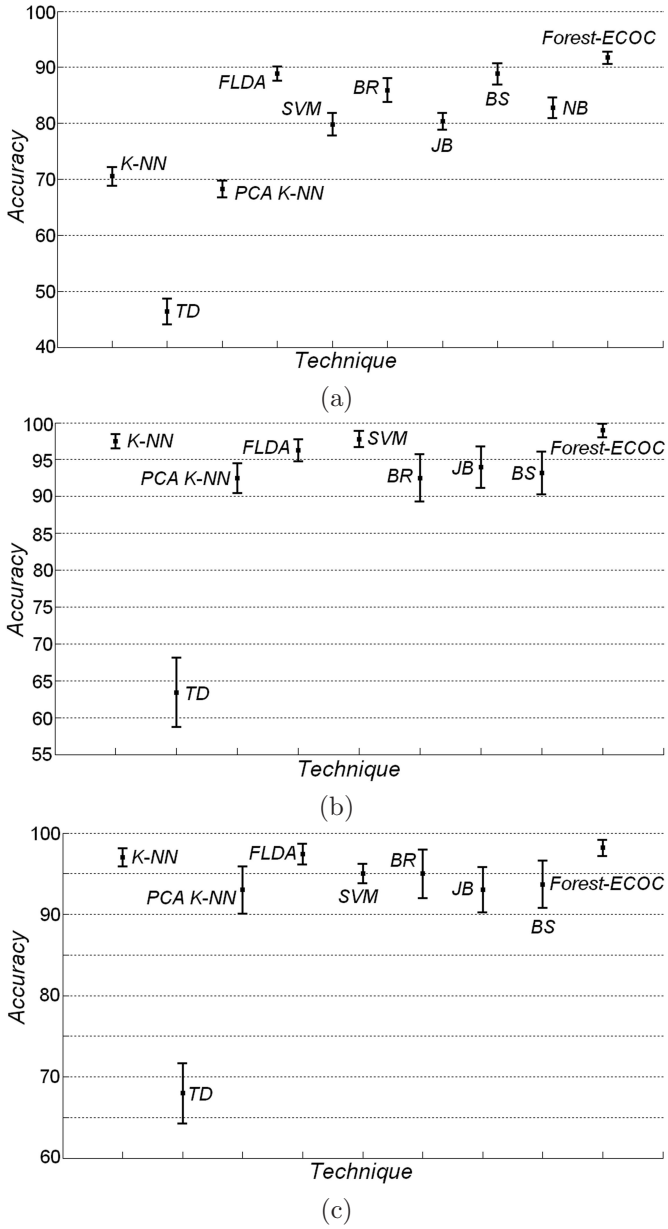


Figure 7.15: Classification results for the (a) Speed, (b) Circular, and (c) Triangular problems.

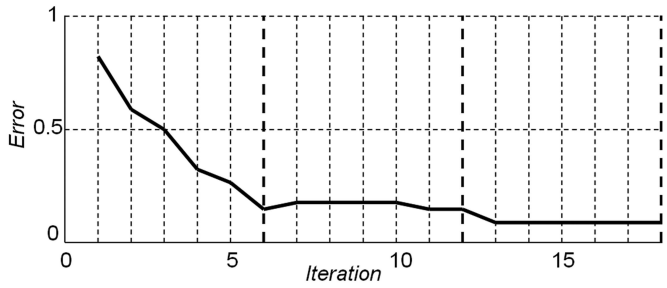


Figure 7.16: Training process of Forest-ECOC embedding the first three optimal trees for the speed group.

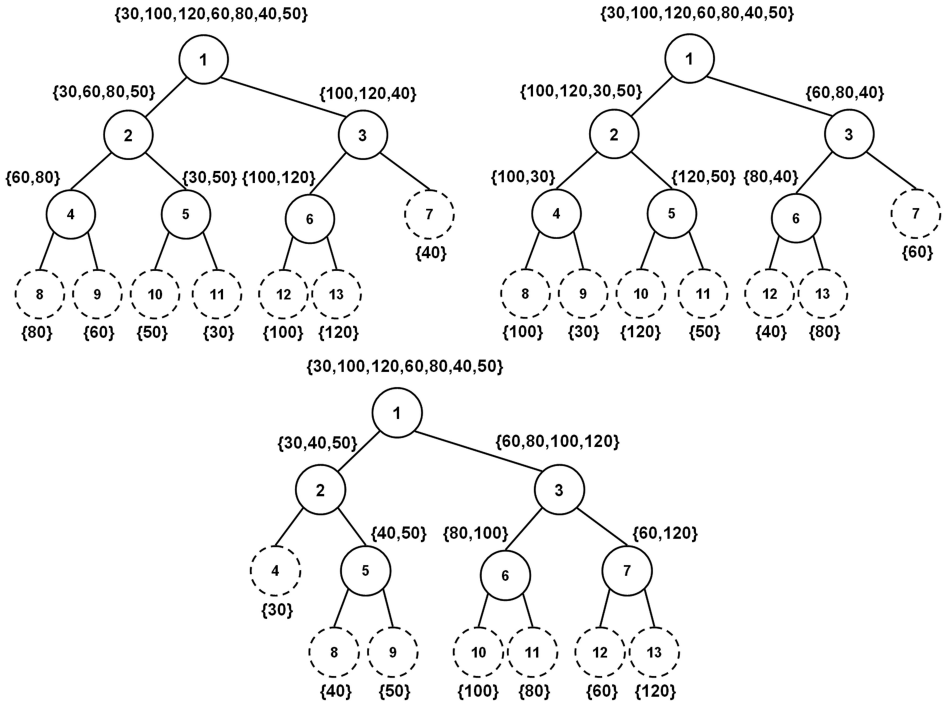


Figure 7.17: Three optimal trees generated by the Forest-ECOC for the speed group.

Figure 7.18: Speed data set samples.

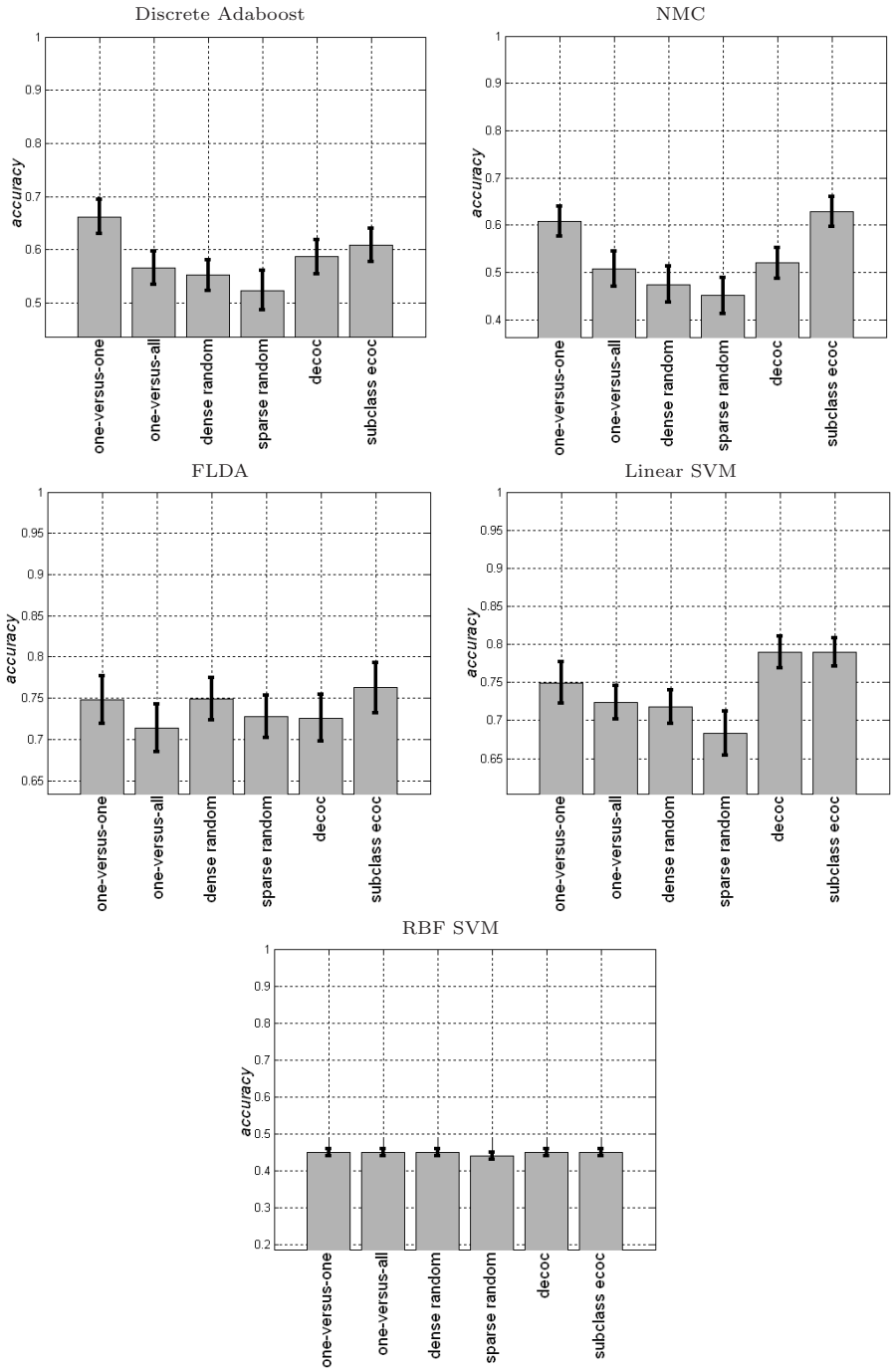


Figure 7.19: Speed data set performances.

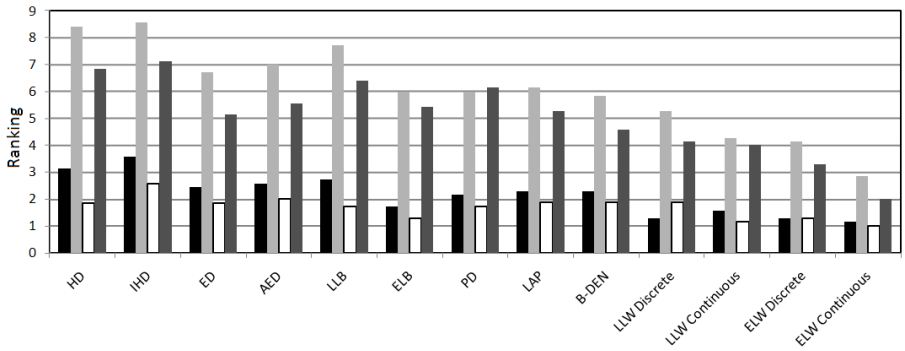


Figure 7.20: Ranking of the decoding strategies for the different coding designs applied over the speed data set: Gentle Adaboost considering (in black) and without (in light grey) considering the intersection of the confidence intervals, and Linear SVM considering (in white) and without considering (in dark grey) the intersection of the confidence intervals, respectively.

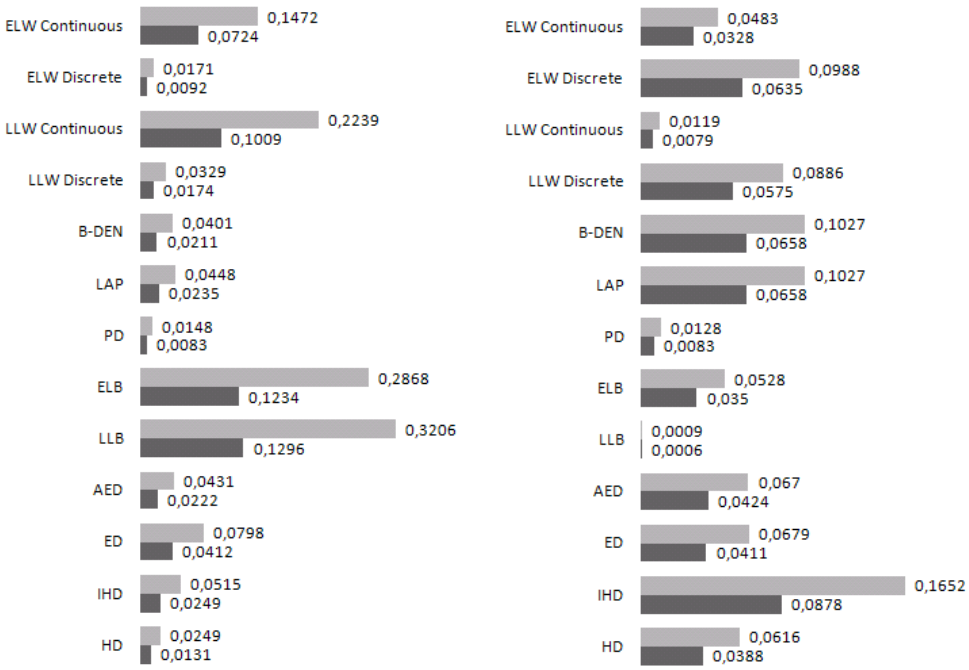


Figure 7.21: Absolute (light lines) and relative (dark lines) improvement for the Sparse Random designs using ternary distance maximization for Gentle Adaboost (left) and Linear SVM (right) on the Traffic sign categorization experiment, respectively.

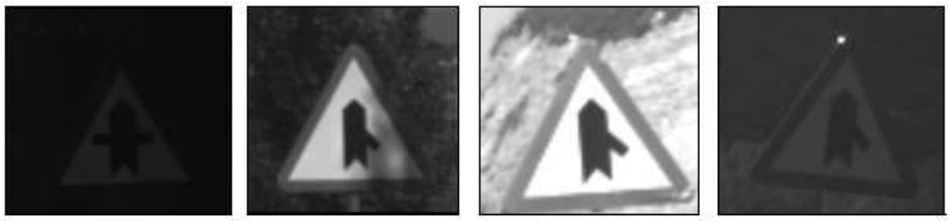
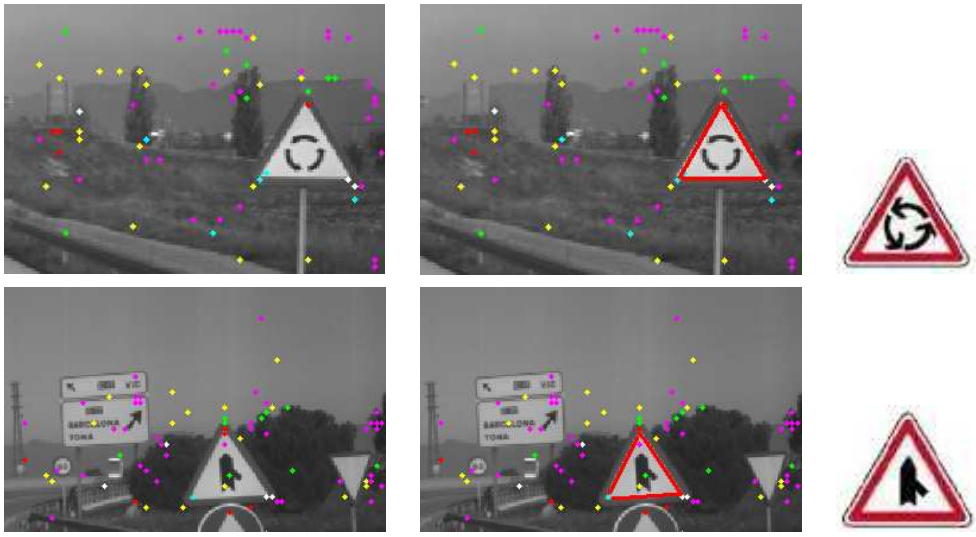


Figure 7.22: Real triangular sign images in non-controlled conditions.



(a) (b) (c)

Figure 7.23: Two examples of the whole procedure for real traffic sign images. (a) Landmark candidates for test images. (b) Predominant likelihoods of landmark combination. (c) Classification results. (landmarks candidates are shown in color).

7.4 Caltech repository data set

At the previous experiments, we showed problems where we performed multi-class categorization. However, in most cases, a previous object detection procedure stage is required before final classification. In this sense we use the Boosted Landmarks of Contextual Descriptors, previously presented in section 6.2, where both parts of the objects and their spatial arrangement are learnt at same time.

7.4.1 Boosted Landmarks in Contextual Descriptors Evaluation

In order to compare the accuracy of our detector, we test the Boosted Landmarks of Contextual Descriptors approach on the Caltech data set [2] considering the following 7 object categories: car side, face, motorbike, car rear, plane, leaf, and spotted cat (fig. 7.24), training only three landmarks from the models of each data set. In fig. 7.25 and fig. 7.26, the models, contour points, landmarks trained, and a correlogram for the face and car side data sets are shown. To validate the method we used 20% of samples to train landmarks (between 30 and 80 samples for each category) and contextual descriptors by boosting, and the rest to test. From the 20% images for training, we select only three representative landmarks in a supervised way to train each data set (fig. 7.24 (down)). We use 40 weaks of Gentle Adaboost with Decision Stumps to train the cascades and the correlogram descriptors.

The results of this experiment are shown in table 7.7. We compare the results with those reported by Fergus et. al. [76] and the boosting context proposed by Amores et. al. [7]. We can see that our proposed technique obtains better results in most of the cases: car (side), face, car (rear), and leaf, and comparable results in the other three cases.

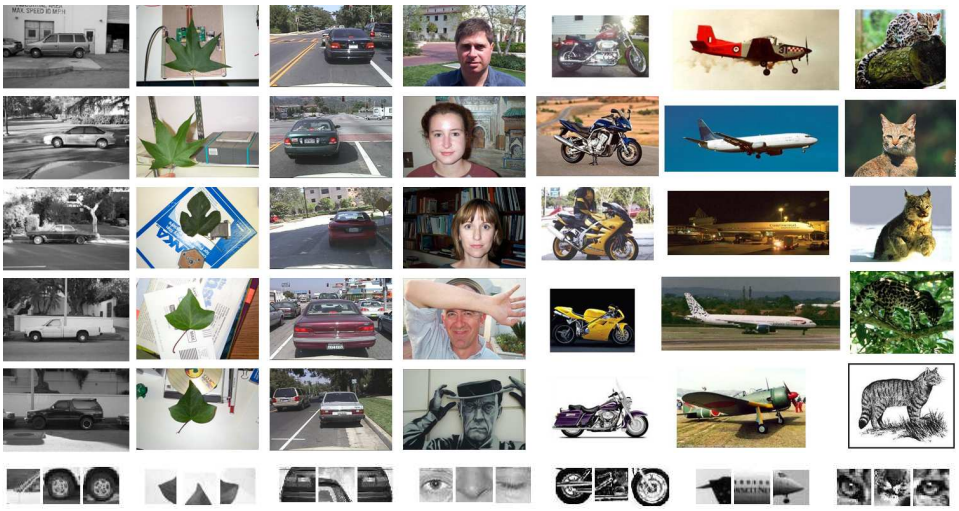
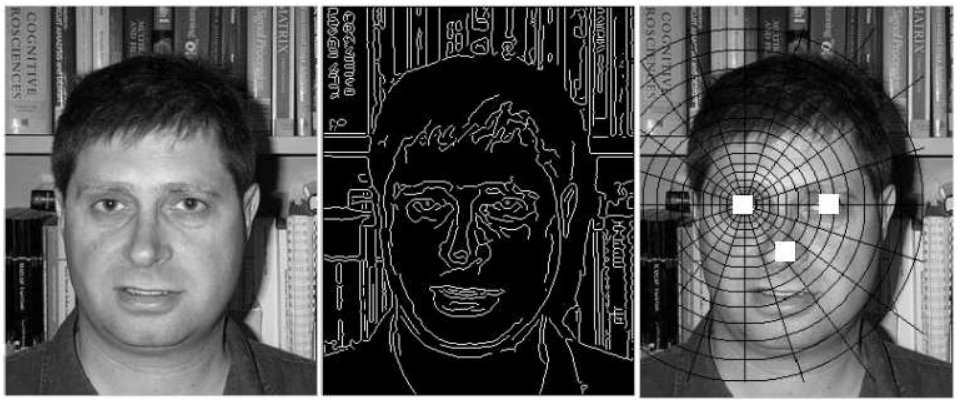


Figure 7.24: Some samples for the considered Caltech categories and relevant landmarks trained.



(a) (b) (c)

Figure 7.25: Fergus faces data set. (a) Original image. (b) Contour points map. (c) Correlogram for a given landmark.

(a) (b) (c)

Figure 7.26: Fergus car side data set (a) Original image. (b) Contour points map. (c) Correlogram of a landmark.

Boosting Context [7] shows very good behavior too, but it is more susceptible to confusion and appearance of false positives and negatives due to the use of the contour points. The authors in [76] use a model that involves a considerable number of features, being susceptible to false positives appearance. We also tested the false alarm rate using the background set of images from the Caltech data set. Testing with 500 background images, our boosted landmarks classifiers obtained a maximum on only one false positive at each of the 7 object categories.

Table 7.7: Hit ratio results for the Fergus data set.

Category	Fergus [76]	Boosting Context [76]	Boosted Landmarks in Contextual Descriptors
Car (side)	88.50%	90.00%	**96.63%**
Face	96.40%	89.50%	**97.72%**
Motorbike	92.50%	**95.00%**	93.85%
car (rear)	90.30%	96.90%	**99.35%**
Plane	90.20%	**94.50%**	92.50%
Leaf	-	96.30%	**98.85%**
Spotted car	**90.00%**	86.50%	84.00%
Rank	2.50	1.86	**1.57**

7.5 Symbol Recognition

Symbol recognition is one of the central topics of Graphics Recognition [47]. A lot of effort has been made in the last decade to develop good symbol and shape recognition methods inspired in either structural or statistic pattern recognition approaches. The presence of handwritten symbols increases the difficulty of classification: there is a high variability in writing style, with different sizes, shapes and intensities, increasing the number of touching and broken symbols. In addition, working with old documents even increases the difficulties in these stages because of paper degradation and the frequent lack of a standard notation. Moreover, due to the fact that architectural, cartographic and musical documents use their own alphabets of symbols (corresponding to the domain-dependent graphic notations used in these documents), the automatic interpretation of such documents requires specific processes.

Two major focus of interest can be stated to deal with symbol recognition problems: the definition of expressive and compact shape description signatures, and the formulation of robust classification methods according to such descriptors. Zhang [97] reviews the main techniques used in this field, mainly classified in contour-based descriptors (i.e. polygonal approximations, chain code, shape signature, and curvature scale space) and region-based descriptors (i.e. Zernike moments, ART, and Legendre moments [52]). A good shape descriptor should guarantee inter-class compactness and intra-class separability, even when describing noisy and distorted shapes. It has been proved that some descriptors, robust with some affine transformations and occlusions in printed symbols, are not efficient enough for handwritten symbols. Thus, the research of other descriptors for elastic and non-uniform distortions is required, coping with variations in writing style and blurring.

The symbol recognition problem is a clear example where the use of a rich descriptor has a determinant effect on the classification performance of object classes. In this section, we perform the classification of real multi-class data sets from different symbol recognition domains. Some examples of each data set are shown in fig. 7.27 and fig. 7.28. Now, we briefly describe each of these data sets:

• **Clefs and alterations data set**: The data set of clefs is obtained from a collection of modern and old musical scores (19th century) of the Archive of the Seminar of Barcelona. The data set contains a total of 3980 samples between the seven different types of clefs from 24 different authors. The models for each of the seven classes are shown in fig. 7.27(a). The images have been obtained from original image documents using a semi-supervised segmentation approach [31]. The main difficulty of this database is the lack of a clear class separability because of the variation of writer styles and the absence of a standard notation.

• **MPEG data set**: The MPEG repository data set [1] has been chosen since it provides a high intra-class variability in terms of scale, rotation, rigid and elastic deformations, as well as a low inter-class variability. A pair of samples for some of the 70 binary object categories are shown in fig. 7.27(b). Each of the classes contains 20 instances, which represents a total of 1400 object samples.

• **Architectural hand-drawn data set**: The architectural symbol database is a benchmark data set that has been created with the logitech IO digital pen [48]. This data set, which has been used in a sketch CAD framework [80], is composed of

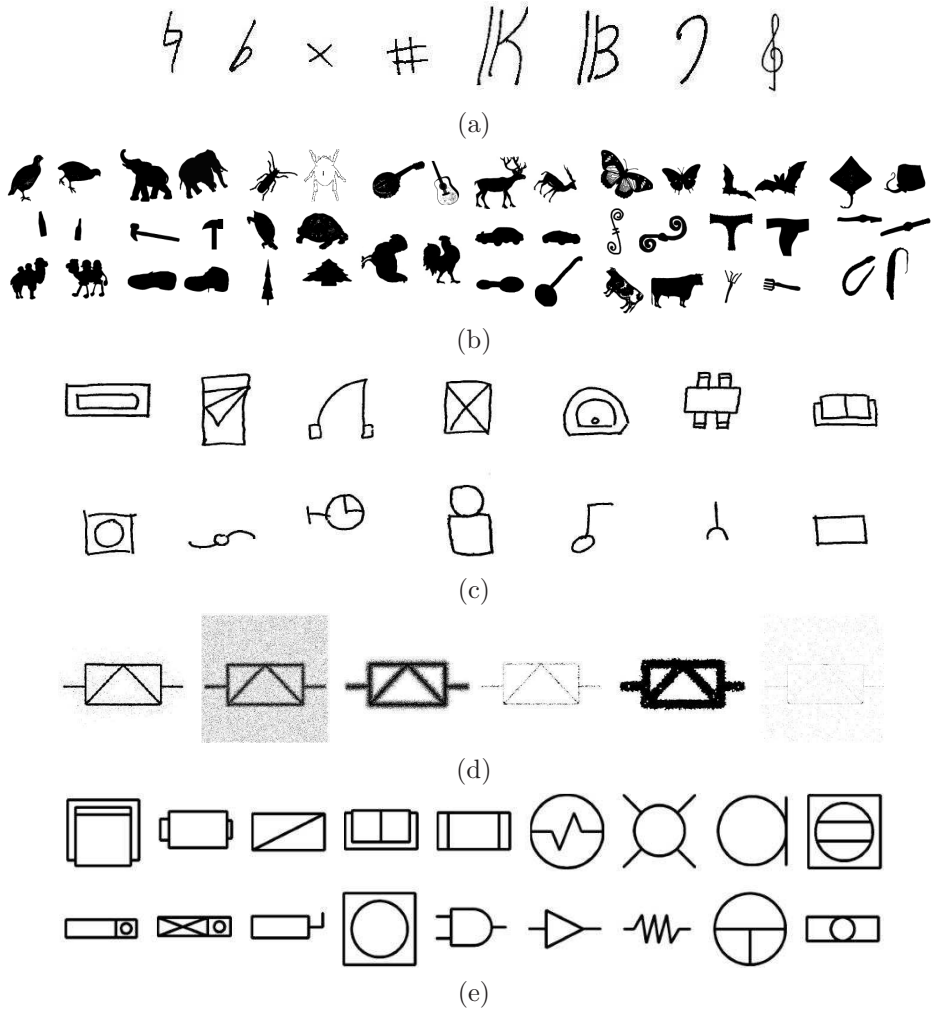
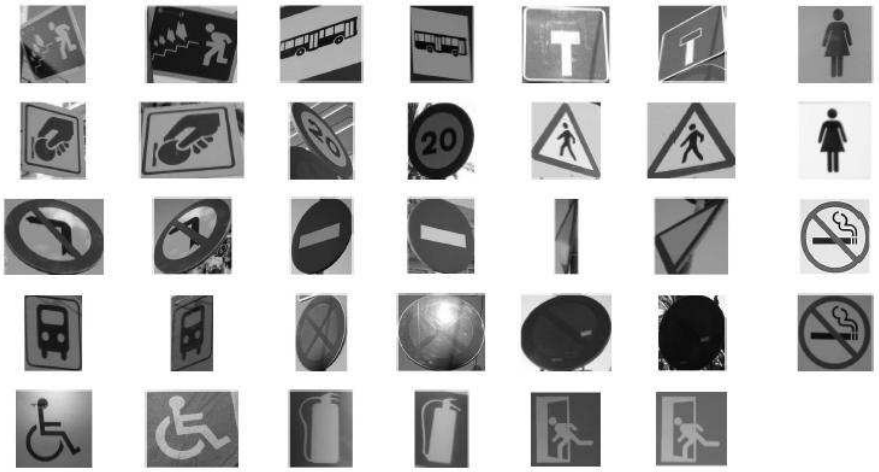


Figure 7.27: Symbol data sets: (a) Clefs and alterations data set, (b) MPEG data set, (c) Architectural hand-drawn data set, (d) GREC05 data set, and (e) GREC07 architectural data set.

(a)



(b)

Figure 7.28: Symbol data sets: (f) GREC07 logos data set and (g) camera-based grey-level symbols data set.

on-line and off-line instances from a set of 50 symbols drawn by a total of 21 users. Each user has drawn a total of 25 symbols and over 11 instances per symbol. The data set consists on more than 5000 instances. To capture the data the following protocol has defined: The authors give to each user a set of 25 dot papers, which are paper containing the special pattern from anoto. Each paper is divided into 24 different spaces where the user has to draw in. The first space is filled with the ideal model of the symbol to guide the users on their draw due to they are not experts on the field of Architectural design. Although the data set is composed of 50 symbols, in our experiments we have chosen the 14 architectural symbols most representative from this database. Our experimental set consists in 2762 total samples organized in the 14 classes shown in fig. 7.27(c). Each class consists of an average of 200 samples drawn by 13 different authors.

• **GREC05 data set**: The GREC2005 database [18] is a public symbol data set with a high number of object categories from architectural and electronic domains shown under different distortions. In particular, we focus on the first level of distortions. Some examples of the used samples are shown in fig. 7.27(d).

• **GREC07 architectural data set**: The GREC2007 data set [19] contains a high number of architectural symbol classes with similar deformations to the GREC05 data set. Some models for the GREC07 architectural classes are shown in fig. 7.27(e).

• **GREC07 logos data set**: The GREC2007 data set [19] also contains a high number of logos different deformations and different levels of distortions. The models for of the logos classes are shown in fig. 7.28(a).

• **Camera-based grey-level symbols data set**: This data set of symbols is composed by grey-level samples from 17 different classes, with a total of 550 samples acquired with a digital camera from real environments. The samples are taken so that there are high affine transformations, partial occlusions, background influence, and high illumination changes. A pair of samples for each of the 17 classes are shown in fig. 7.28(b).

The previous data sets of symbol recognition problems are highly affected by many types of deformations, such as: intra-class and inter-class variabilities, elastic and rigid deformations, rotations, occlusions, changes in the point of view, different writing styles, etc. The main objective of this section is to show the influence of a rich descriptor to obtain a high generalization capability of the classification methodology applied. For this task, we fixed the classification strategy using standard ECOC designs and state-of-the-art base classifiers to compare the effectiveness of different types of feature sets with our Blurred Shape Model descriptor presented in section 6.1.

7.5.1 Clefs and alterations data set classification

For the classification of the clefs and alterations data sets, our BSM descriptor is compared with ART, Zoning, Zernike, and CSS curvature descriptors from the standard MPEG [97][43][57]. Moreover, we compare with the SIFT descriptor, which has shown to be a dominant strategy applied on real description problems. For all the experiments, stratified ten-fold cross-validation with a two-tailed t-test at 95% of the confidence interval is used. The descriptors for BSM and Zoning techniques are of length 8×8 from the considered sub-regions. The parameters for ART are radial

order with value 2 and angular order with value 11. For the Zernike descriptor, 7 Zernike moments are used. And a length of 200 with an initial sigma of 1 increasing per one is applied for the curvature space of the CSS descriptor. The ECOC design applied is one-versus-one with Exponential Loss-Weighted decoding. Different base classifiers are applied: $FLDA$, Linear SVM, $RBF\ SVM$, and Discrete Adaboost. The classification performance considering the different descriptors and classification strategies are shown in table 7.8. One can see that the BSM descriptor is more robust against the elastic deformations produced by the writing styles, and independently of the classification methodology applied, the BSM features tends to outperform the rest of feature sets. In this particular domain, the best results are obtaining using Discrete Adaboost as the base classifier.

Table 7.8: Clefs and alterations classification performances.

	$FLDA$	Linear SVM	$RBF\ SVM$	Discrete Adaboost
BSM	83.53(7.52)	79.45(6.30)	80.43(6.17)	**88.99(5.00)**
Zoning	78.62(7.28)	80.51(7.31)	81.54(7.52)	**83.61(5.24)**
SIFT	71.35(9.04)	**76.45(6.73)**	75.47(9.76)	74.95(9.77)
CSS	68.76(11.02)	66.87(8.19)	69.87(9.18)	**71.33(8.44)**
Zernike	69.09(6.01)	71.66(8.29)	39.21(9.00)	**72.05(7.76)**

7.5.2 MPEG data set classification

Considering the MPEG data set, we perform two types of experiments. First, we use the 23 categories from the MPEG repository database shown in fig. 7.27(b). This sub set from the 70 original MPEG data set classes has been chosen since it provides a high intra-class variability in terms of scale, rotation, rigid and elastic deformations, as well as a low inter-class variability. Each of the classes contains 20 instances, which represents a total of 460 object samples for the first experiment. The details of the descriptors are the same than at the previous experiment.

For the first experiment, we started the classification using the first three classes of fig. 7.27(b). Iteratively, one class is added at each step, and the classification is repeated until the 23 classes are processed. The main objective is to analyze the performance of the techniques when the number of classes increases. The results of the experiment are shown in fig. 7.29. Observing the figure, one can realize that the BSM descriptor attains the best performance for any number of classes in the classification system. Besides, an important point is that its performance does not decrease significatively while increasing the number of classes, obtaining results around 80% in all cases. The second descriptor in the ranking is Zernike, which offers similar performance than BSM when the number of classes is small, but substantially decreases with the number of object categories. Finally, Zoning, CSS, and ART descriptors offer the worst classification scores in this problem. This can be intuitively justified by the fact that Zoning descriptors are very local, and the database is full of shape variations. This fact also affects to the CSS descriptor, since the points of curvature varies due to the high shape variations among objects.

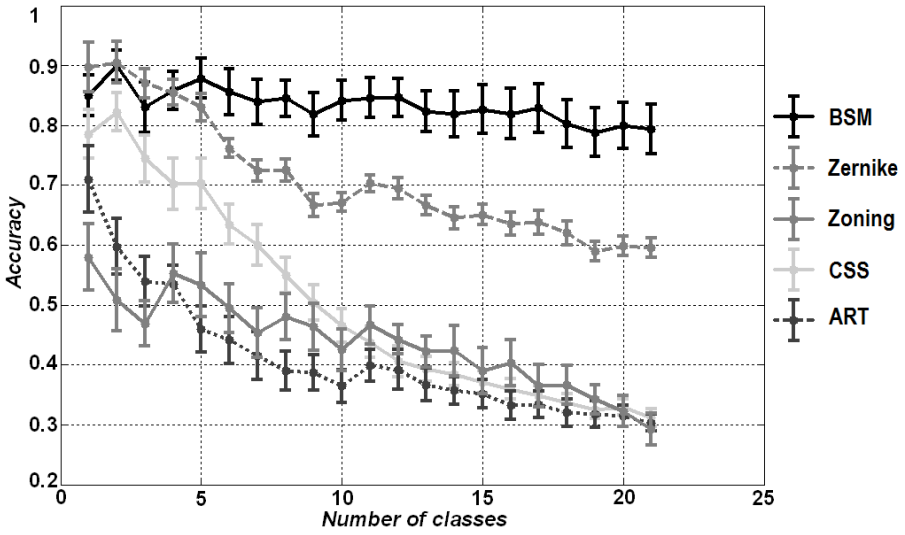


Figure 7.29: Classification of MPEG data set for different number of classes and descriptor types.

Table 7.9: Classification accuracy on the 70 MPEG7 object categories for the different descriptors using 3-Nearest Neighbor and our system.

Descriptor	$3-NN$	one-versus-one ECOC LW
BSM	**0.6579±0.1203**	**0.7793±0.0725**
Zernique	0.4364±0.0766	0.5129±0.0548
Zoning	0.6064±0.1197	0.6550±0.0664
CSS	0.3701±0.1076	0.4454±0.0711
ART	0.2443±0.0169	0.2873±0.0646
SIFT	0.2914±0.0568	0.3257±0.0404

The second experiment consists of classifying the whole set of 70 MPEG7 classes. The classification results are shown in table 7.9 using a 3-NN classifier and a one-versus-one design with Discrete Adaboost as the base classifier and Exponential Loss-Weighted decoding. Note that the use of an ECOC scheme considerable increase the classification performance, and that the BSM features allow a final performance near 80% classifying the set of 70 MPEG classes.

7.5.3 GREC05 classification

Our tests on the GREC2005 database are applied on the first level of distortions. We have generated 140 artificial images per model (thus, for each of the 25 classes) applying different distortions such as morphological operations, noise addition, and partial occlusions. Then, we use a one-versus-one design with Discrete Adaboost as the base classifier and Exponential Loss-Weighted decoding to learn the synthetic data

Method	Distortion Level 1	Distortion Level 2	Distortion Level 3	Distortion Level 4	Distortion Level 5	Distortion Level 6
KDM	100	100	100	96	88	76
BSM	100	100	100	100	96	92

Figure 7.30: Descriptors classification accuracy increasing the distortion level of GREC05 database using 25 models and 50 test images.

Table 7.10: Architectural GREC07 contest tests performed.

Test	Kind	Model Image	Test Images	Rotation	Scaling	Degradation
5	Structured	50	200	None	Random	Random

for the different classes. The BSM grid size in this case is of 28×28 bins after looking for the optimum grid using applying cross-validation. In this sense, 784 features are extracted from every image, from which Adaboost selects a maximum of 50.

For this experiment, the classification is performed over 25 classes of different distortion levels from the GREC05 data set. We compare our results with the ones reported in [98] using the kernel density matching method (KDM). The results are shown in fig. 7.30. One can see that the performances obtained with our methodology are very promising, outperforming for some levels of distortions the KDM results.

7.5.4 GREC07 architectural data set classification

We used the BSM desciptor in a one-versus-one design with Discrete Adaboost as the base classifier and Exponential Loss-Weighted decoding to learn the test described in table 7.10 from the GREC07 architectural data set. The performance obtained on this public data set is of 91.5%. Note that this percentage is high considering the high number of classes and the degree of degradation of the samples.

7.5.5 GREC07 logos data set classification

We used the BSM desciptor in a one-versus-one design with Discrete Adaboost as the base classifier and Exponential Loss-Weighted decoding to learn the tests described in table 7.11 from the GREC07 logos data set. The performance obtained on this public data sets are shown in table 7.12. Note that these percentages are high considering the high number of classes and the degree of degradation of the samples involved at each experiment.

Table 7.11: Logos GREC07 contest tests performed.

Test	Kind	Model Image	Test Images	Rotation	Scaling	Degradation
8	Logos	105	300	None	Random	None
10	Logos	105	300	None	None	Random
11	Logos	105	200	None	None	Random
12	Logos	105	300	None	None	Random

Table 7.12: Logos GREC07 data set results.

Test	Performance
#8	95%
#10	82.7%
#11	56.5%
#12	55%

7.5.6 Camera-based grey-level symbols data set

For the type of images of this data set the SIFT descriptor has demonstrated to be a dominant strategy [49]. In this sense, for this experiment we compare the BSM and SIFT descriptors.

To extend the use of the BSM descriptor from binary to grey-scale images, we estimate an adaptive orientation threshold for each particular problem. For a given image, our method computes the gradient module and normalizes it to unit. Then, the histogram of gradient magnitudes is estimated, and the Otsu method is applied in order to obtain an adaptive threshold for significant gradient modules. The points in the image with a higher gradient module than the computed threshold use to correspond to relevant symbol shape points. Some examples of the data set of this experiment and their corresponding BSM descriptors are shown in Fig. 7.31.

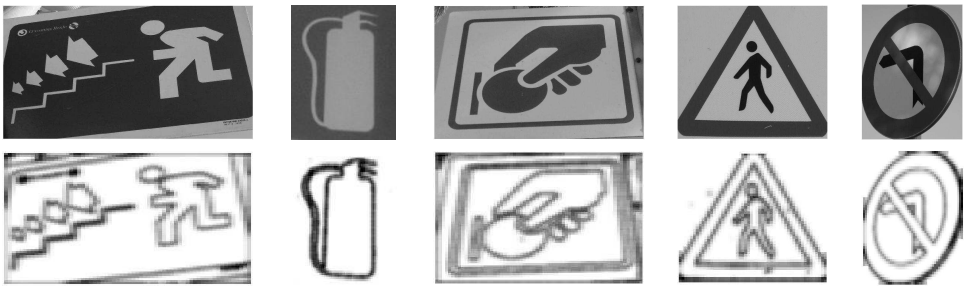


Figure 7.31: BSM descriptors from samples of the grey-level symbols data set.

The performance and confidence interval obtained in this experiment from a ten-fold cross-validation using the BSM and SIFT descriptors in a one-versus-one ECOC scheme with Gentle Adaboost as the base classifier is shown in table 7.13. One can see that the result obtained by the BSM descriptor adapted to grey-scale symbols significantly outperforms the result obtained by the SIFT descriptor. This difference is produced in this data set because of the high changes in the point of view of the symbols and the background influence, which produce significant changes of the SIFT orientations.

BSM	SIFT
75.23(7.18)	62.12(9.08)

Table 7.13: Performance of the BSM and SIFT descriptors on the grey-scale symbols data set using a one-versus-one ECOC scheme with Gentle Adaboost as the base classifier.

7.6 Applications discussion

In this Chapter, we presented different real and synthetic multi-class problems where we tested our methodology. First, we modelled a multi-class medical image problem: Intravascular Ultrasound Tissue characterization. We used the different ECOC configurations presented in this thesis to model the problem and to compare different state-of-the-art strategies. Second, we characterized the level of damage of patients with the Chaga's disease. In this case, the decoding methodology was evaluated and successfully applied, outperforming related works treating the same data. Third, we presented a Mobile Mapping System and we proposed a methodology based on ECOC to detect and classify a wide set of traffic sign classes. Fourth, we use the previous Boosted Landmarks of Contextual descriptors to model those problems where a previous object detection is required before applying classification. The detection methodology was evaluated on some categories from the public Caltech repository data set and at the previous Mobile Mapping System. Finally, we modelled several data sets from the symbol recognition domain. These data sets are affected by degradation on the shape of the object. In this sense, we used the previous Blurred Shape Model methodology to describe this type of data, which successfully describe symbols with high level of degradation of the shape.

Chapter 8

Conclusion

Real-life situations are full of multi-class classification tasks. In this sense, **Error-Correcting Output Codes** demonstrated to be a powerful tool to model the data provided by these multi-class situations. In this thesis, we presented an Error-Correcting Output Coding methodology to deal with multi-class categorization problems. As we showed in this thesis, there still exist some drawbacks in the definition of the ECOC framework that stop down the reliability of the ECOC designs.

8.1 Summary and contribution

The common way to model a multi-class problem using an ECOC design is by means of a coding and a decoding strategy:

1) We presented a novel problem-dependent methodology to deal with the **coding step** of an Error-Correcting Output Codes design. The state-of-the-art ECOC coding designs are problem-independent. It means that to obtain an ECOC system with a good generalization performance, large codes are required. This implies a high computational cost for learning and testing the system. At the same time, problem-independent designs can not assure that the learnt boundaries based on the binary problems are the optimal ones for a given problem. These drawbacks motivated us the design of problem-dependent ECOC strategies.

Three alternatives to design a problem-dependent ECOC matrix were proposed. The common point for all the strategies in that all of them exploit the problem-domain to obtain small codewords with high discriminative power.

1.1) The **Forest-ECOC** strategy takes advantage of a forest of sub-optimal binary tree structures. Each internal node of the trees is embedded as a binary problem in an ECOC coding matrix. In this sense, we can guarantee a high generalization performance with a small number of embedded tree structures. Moreover, the main advantage of this representation is that all the information provided by the trees for a particular class are taken into account jointly in order to obtain a classification decision. On the other hand, the criterion for the design of the tree structures is an important point, which decides the generalization capability of the designed multi-class system. Depending on the problem we are working on, greedy searches and

sub-optimal entropy-based solutions are proposed to design binary tree structures. This method proposes a sub-optimal solution to a multi-class problem because of the sub-optimal designs of the tree structures.

1.2) One of the weak points of the Forest representation for ECOC designs is that at each iteration of the procedure a whole tree structure is embedded in the ECOC matrix, though some of the internal nodes would not be necessary to improve the knowledge of the system. Concerning to this problem, the second problem-dependent proposal, the Optimizing Node Embedding (**ECOC-ONE**), evaluates at each step of the algorithm the generalization of the system over a training and a validation sub-sets. Then, a new binary problem that learns the classes that decrease the performance of the system is embedded to update the coding matrix. In this case, the performance of the ECOC design is evaluated each time that a new binary problem is added in the system, and the process is repeated meanwhile the performance of the system improves or the training error is under a certain epsilon value. This problem-dependent design showed to obtain robust results learning different types of multi-class data, obtaining better performance than traditional ECOC techniques using far less number of binary problems.

1.3) Finally, the last problem-dependent design presented in this thesis is the **Sub-class ECOC strategy**. Although the previous designs propose an advanced design of Error-Correcting Output Codes with high generalization capability, the performance of the whole system is still determined by the ability of the base classifier to learn each binary problem. In this sense, the Sub-class ECOC approach avoids the limitations of the rest of ECOC designs when some distributions of the data are difficult to model using some types of base classifiers due to the overlapping of the data. The Sub-class strategy splits classes into sub-classes based on a clustering approach for the cases that the base classifier is not capable to distinguish the classes. Sequential Forward Floating Search based on maximizing the Mutual Information is used to generate the sub-groups of problems that are split into more simple ones until the base classifier is able to learn the original problem. In this way, multi-class problems which can not be modeled by using the original set of classes are modeled without the need of using more complex classifiers. The final ECOC design is obtained by combining the sub-problems. As a consequence of applying the Sub-class strategy, in the worst case it remains the same than without using sub-classes. One of the important points is that both, base classifier and sub-class, can be optimized. If the base classifier is well tuned, less binary problems and sub-classes would be required by the Sub-class strategy. On the other hand, the Sub-class approach could be seen as an incremental tool independent of the base classifier to improve the weakness of the base classifiers. Moreover, the sub-class scheme not only can be used in the present methodology, but also to improve other multi-class strategies.

2) We also analyzed the behavior of the state-of-the-art **decoding strategies** applied over 3-symbol coding matrices. As shown on the ECOC coding chapter, the ECOC designs that attains the best performances are 3-symbol based. The main reason is that the binary ECOC is a particular case of the ternary ECOC framework. The use of the third symbol allows us to design a more rich set of binary problems to adapt the ECOC design to each particular problem domain. However, the decoding strategies presented in the literature that are applied over 3-symbol ECOC matrices

were designed to deal with just two symbols. In this sense, we showed some inconsistencies produced by the traditional decoding strategies when using the zero symbol. We presented two working hypotheses to deal with a successful decoding and analyzed the decoding strategies over a new **taxonomy of decoding strategies**. As a consequence, different strategies fulfilling the presented properties were proposed:

2.1) Attenuated Euclidean decoding: This technique is proposed to avoid the influence of the ECOC coding matrix positions that do not provide relevant information of the data.

2.2) Laplacian decoding: This technique introduces a measure that counts the number of coincidences between the input codeword and the class codeword, normalizing by the total number of codeword positions. The procedure introduces a previous bias to make the technique robust in cases of having a small number of coded positions in one word.

2.3) Pessimistic β-density decoding: This technique estimates the probability density functions between two codewords. The main goal of this strategy is to model at the same time the accuracy and uncertainty based on a pessimistic score on the continuous binomial distribution in order to obtain more reliable predictions.

2.4) Loss-Weighted decoding: The Loss-Weighted decoding strategy codifies a matrix of weights that ponders the decoding process. This matrix avoids the influence of the positions that do not provided information at the coding step. At same time, the technique makes the decoding measures between codewords comparable either in the binary as in the ternary ECOC framework.

We showed that when the new decoding strategies avoid the *bias* produced by the zero symbol and all the codewords work in the same *dynamic range*, the new strategies significantly outperform the traditional decoding methodologies independently of the coding strategy applied. In particular, we showed that the performance improvements are more significant when the new rules are applied over coding designs with higher sparseness degree (high percentage of zero symbols in the coding matrix). It is produced because when we increase the percentage of zero symbols, the two biases produced by the third symbol also increase, and the classification performance for the traditional decoding strategies is more affected.

From the comparison among the presented decoding strategies, we found that the least improvement is achieved by the Attenuated Euclidean decoding, followed by the Laplacian, the Pessimistic β-density decoding, and finally, the Loss-Weighted decoding. This order corresponds to a sort from discrete to continuous approaches, taking into account that the last ones are also sorted based on the amount of continuous information that votes the decoding process. Thus, the order also corresponds to the level of complexity of the decoding strategies.

3) Moreover, we showed that the ternary ECOC framework contains inconsistences not only at the decoding step, but also in the definition of Sparse coding matrices. We showed that the rows separability in terms of the Hamming distance of the binary ECOC framework can not be applied in the ternary case, and we presented a new **formulation of the ternary ECOC distance and error-correcting capabilities in the ternary ECOC framework**. Based on the new measure, we stress on how to design **coding matrices preventing coding ambiguity** and propose a new **Sparse Random coding matrix with ternary distance maximization**. Comparing the

results of the traditional and novel strategies of Sparse random matrix generation, we found that the performance improvements obtained by the new methodology are statistically significant.

4) Furthermore, we introduced two new techniques for object detection and description that can be used jointly with the multi-class ECOC methodology.

4.1) First, the **Blurred Shape Model** technique describes the content of a region by considering the relevant gradient magnitude points to define a probability density map of the shape of the object, even if it suffers from irregular deformations.

4.2) And second, we introduced the **Boosted Landmarks of Contextual Descriptors**, a new object detection method based on training the discriminant features of the object description. Such description includes the information of correlograms to learn at the same time the object local representation and the spatial relationship among its parts fragments.

5) The multi-class methodology presented in this thesis has been compared with the state-of-the-art ECOC designs, multi-class classifiers, and object detection and description strategies. Statistical tests have been performed to look for statistical significance among method performances. Many real and synthetic multi-class data sets have been used to evaluate the methodology, such as the multi-class data sets from the public **UCI Machine Learning Repository**. We presented a set of **real applications** were our methodology is applied and compared with the state-of-the-art strategies. **We modelled an Intravascular Ultrasound tissue characterization problem, the categorization of the level of coronary damage of patients with the Chaga's disease, a multi-class traffic sign classification problem from a Mobile Mapping System, and a wide set of real and synthetic multi-class benchmarking and symbol recognition data sets**.

As a conclusion, we can state that in this thesis we defined a new methodology that exploits the domain of each particular multi-class problem to codify a problem-dependent ECOC coding matrix. New decoding strategies are applied to take full-benefit from the information provided at the coding step, and significant performance improvements are obtained compared to the state-of-the-art ECOC designs and multi-class classifiers. The techniques obtain high generalization performance with a small code length, being combined with object detection and description strategies and solving several multi-class real world problems.

8.2 Future work

As future lines, it would be interesting to analyze how different the ECOC problem-dependent designs evolve depending on the base classifier that we consider. In the same way, we could take into account a wider set of tuned multi-class built-in classifiers to compare with the benefits of the new ECOC designs. Another possible extension consists on the inclusion of continuous information in the construction of the problem-dependent ECOC matrix.

An interesting analysis on the ECOC design could be its relation with the multi-task framework. The way in which an ECOC codeword takes into account simultaneously the responses of all the dichotomizers can be seen as a multi-task problem [15].

In this sense, a further analysis to develop a problem-dependent multi-task ECOC could benefit from both frameworks.

Particularly, in the case of the proposed problem-dependent ECOC designs, an open issue is the selection of a proper solution to guide the forest-ECOC matrix construction. In the case of the ECOC-ONE procedure, it could benefit from a further analysis of the convergence procedure of the matrix construction.

In the case of the sub-class ECOC approach, a future line of work is to use the sub-classes obtained by the splitting procedure to construct other types of problem-dependent ECOC matrices. For example, the final sub-classes can be potentially useful to improve other multi-class strategies or standard hierarchical clustering strategy. Note that the final set of sub-classes can be used for example over the one-versus-one ECOC configuration. Obviously, it will require a higher number of dichotomizers to codify the problem, but in cases where the computational cost is not a problem, it could result a promising choice.

Moreover, a further analysis on the clustering strategy applied on the Sub-class approach, based on the behavior of each particular base classifier, can help the convergence of the algorithm to model multi-class problems minimizing the number of required sub-classes.

In the case of the new Sparse random design, an open issue is to construct the random ECOC matrix using an alternative ternary distance maximization, such as for example based on the ternary decoding measures proposed in this thesis.

In the case of the visual pattern recognition problems, further analysis is required to develop techniques robust to a wide range of image/object transformations. The research of a generic object recognition methodology is still an open issue, and several researches are working on this problem. We want to take full advantage of the state-of-the-art object recognition procedures (object detection methods in most cases) to combine their behavior with our multi-class methodology in order to develop a robust multi-class object recognition procedure.

From a practical point of view, we are developing a public ECOC toolbox that can be useful to other authors for replicating experiments and comparing different methodologies.

Appendix A

ECOC Notation

Table A.1: ECOC Notation.

Δ - Matrix composed by the Hamming distances between the codewords of M

ρ^j - j^{th} feature of the object (data sample) ρ

ν_t - Confusion matrix of the training data

ν_v - Confusion matrix of the validation data

$\theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$ - Parameters for the size, performance, and improvement.

ζ_j - Centroid of cluster j

ξ - Error function

α - Number of matches

β - Number of failures

φ - Conditional function

$\psi(\nu) = [\psi_1(\nu), \psi_2(\nu), ..., \psi_N(\nu)]$ - Set of Beta Density distributions for N classes

a - Value introduced by a match in a position coded by $\{-1, +1\}$

a_t - Training accuracy

a_v - Validation accuracy

b - Error induced by the zero symbol

CD - Critical difference for the Nemenyi test

C - Set of classes

c_i - Class i

$d(y_{i_1}, y_{i_1})$ - Decoding measure between codewords of classes c_{i_1} and c_{i_2}

d - Distance

d_c - Minimum Hamming distance between all pairs of columns of M and their opposites

d_r - Minimum Hamming distance among all pairs of codewords

d_t - Ternary distance

$D = [d_1, ..., d_N]$ - Vector of distances

e - Value introduced by a failure in a position coded by $\{-1, +1\}$

e - Euler number (or Neper constant)

F_F - Corrected Friedman Statistic

$\{f_1, ..., f_n\}$ - Set of continuous hypotheses, $f_j \in R$

H - Matrix of accuracy of hypotheses

$\{h_1, ..., h_n\}$ - Discrete hypotheses set, $h_j \in \{+1, -1\}$

I_b, I_a, I_e - Sets of coordinates of a codeword corresponding to the zero positions, matches on $\{-1, +1\}$ values, and failures on $\{-1, +1\}$ values, respectively.

I - Mutual information

J - Data matrix of the original problem

J_i - Data of class C_i

Table A.2: ECOC Notation.

J' - Data matrix of the sub-classes

k - Number of methods to compare

K_m - Objective function

K - Number of classes considered by a classifier in the Laplace correction

K_1, K_2 - Constant factors

$\ell = \{\ell_1, .., \ell_N\}$ - Set of labels

$L(\theta)$ - Loss-based function of parameter θ

L - Sets of classes labels

$l(\rho) = \ell_i$ - Label function of data sample ρ is ℓ_i

M_W - Matrix of weights

$M \in \{-1, +1\}^{N \times n}$ - Binary coding matrix

$M \in \{-1, 0, +1\}^{N \times n}$ - Ternary coding matrix

m - Number of objects

N - Number of classes

n - Number of binary problems

$P(X)$ - Probability of item X

P - Prior

p - Probability density function

$\wp_i = \{\wp_i^+, \wp_i^-\}$ - Set of positive and negative sub-sets of the i^{th} binary problem

q_α - Studentized range statistic divided by $\sqrt{2}$

r_i^j - Rank of each problem i and each ECOC design j

R_j - Mean rank of the j^{th} design

S - Element of L

S_t - Training set

S_v - Validation set

s_i - Pessimistic score of class c_i

u - Threshold parameter

υ, ω - Optimization parameters

W - Set of weighting values

w - Weight

X_F^2 - Friedman Statistic

X - Set of objects

x - Test codeword

y_i - Codeword of class c_i

Y_i^e - Extended set of codewords for class c_i z_i - Number of zero symbols of codeword y_i

Appendix B

Sequential Forward Floating Search (*SFFS*)

Table B.1: Sequential Forward Floating Search (SFFS) algorithm.

Input:
$Y = \{x_j | j = 1..D\}$//Avaliable items//

Output:
$X_k = \{x_j | j = 1..|Y| (or D), x_j \in Y\}$

[**Initialization:**]
$X_0 = \{\emptyset\}; k = 0$

[**Termination:**]
Stop when the criterion does not increase $J(X_k) \approx J(X_{k-1})$[a]

Step 1 (**Inclusion**)
$x^+ = argmax_{x \in Y - X_k} J(X_k \cup x)$
$X_{k+1} = X_k \cup x^+, k = k + 1$

Step 2 (**Conditional exclusion**)
$x^- = argmax_{x \in X_k} J(X_k - x)$

if $J(X_k - x^-) > J(X_{k-1})$ then
$X_{k+1} = X_k - x^-, k = k + 1$
go to **Step 2**

else
go to **Step 1**

[a] We apply the Fast Quadratic Mutual Information *MI*, see Appendix I.

The *SFFS* process of table B.1 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. Y is our set of classes to be partitioned. Our discriminability criterion is the mutual information (MI). Our goal is to maximize the MI between the data in the sets and the class labels created for each subset [73].

Appendix C

Fast Quadratic Mutual Information *MI*

Let $\mathbf{x}$ and $\mathbf{y}$ represent two random variables, and let $p(\mathbf{x})$ and $p(\mathbf{y})$ be their respective probability density functions. The mutual information measures the dependence between both variables, and is defined as follows:

$$I(\mathbf{x},\mathbf{y}) = \int\int p(\mathbf{x},\mathbf{y}) \log\left(\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}\right) d\mathbf{x}d\mathbf{y} \tag{C.1}$$

Observe that mutual information is zero if $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$. It is important to note that eq. C.1 can be seen as a Kullback-Leiber divergence, defined in the following way:

$$K(f,g) = \int f(y) \log\left(\frac{f(y)}{g(y)}\right) dy \tag{C.2}$$

where $f(y)$ is replaced with $p(\mathbf{x},\mathbf{y})$ and $g(y)$ with $p(\mathbf{x})p(\mathbf{y})$.

Alternatively, Kapur et al. [39] argued 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 $D(f,g) = \int (f(y) - g(y))^2 dy$. As a result, it was proved that maximizing $K(f,g)$ is equivalent to maximizing $D(f,g)$. Therefore, we can define the quadratic mutual information as follows:

$$I_Q(\mathbf{x},\mathbf{y}) = \int\int (p(\mathbf{x},\mathbf{y}) - p(\mathbf{x})p(\mathbf{y}))^2 d\mathbf{x}d\mathbf{y} \tag{C.3}$$

The estimation of the density functions of I_Q can be done using the Parzen window estimator. In that case, when combined with Gaussian kernels we can use the following property: Let $N(y,\Sigma)$ be a d-dimensional Gaussian function, it can be shown that:

$$\int N(y - a_1, \Sigma_1) N(y - a_2, \Sigma_2) dy = N(a_1 - a_2, \Sigma_1 + \Sigma_2) \tag{C.4}$$

Observe that the use of this property avoids the computation of one integral function.

In particular, we compute the mutual information between the random variable of the features $\mathbf{x}$ and the discrete random variable associated to the class labels created for a given partition ($\mathbf{d}$). The notation for the practical implementation of I_Q is as follows: Assume that we have N samples in the whole data set; J_p are the samples of the class p; N 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 p. Then, $p(\mathbf{d})$ and $p(\mathbf{x}|\mathbf{d})$ can be written as:

$p(\mathbf{d} = p) = \frac{J_p}{N}$

$p(\mathbf{x}|\mathbf{d} = p) = \frac{1}{J_p} \sum_{j=1}^{J_p} N(x - x_{pj}, \sigma^2 I)$

$p(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} N(x - x_j, \sigma^2 I)$

Expanding eq. C.3 and using a Parzen estimate with a symmetrical kernel with width σ, we obtain the following equation:

$$I_Q(\mathbf{x}, \mathbf{d}) = V_{IN} + V_{ALL} - 2V_{BTW} \tag{C.5}$$

where:

$V_{IN} = \sum \int p(\mathbf{x}, \mathbf{d})^2 dx = \frac{1}{N^2} \sum_{p=1}^{N} \sum_{l=1}^{J_p} \sum_{k=1}^{J_p} N(x_{pl} - x_{pk}, 2\sigma^2 I)$

$V_{ALL} = \sum \int p(\mathbf{x})^2 p(\mathbf{d})^2 dx = \frac{1}{N^2} \sum_{p=1}^{N} (\frac{J_p}{N})^2 \sum_{l=1}^{N} \sum_{k=1}^{N} N(x_l - x_k, 2\sigma^2 I)$

$V_{BTW} = \sum \int p(\mathbf{x}, \mathbf{d}) p(\mathbf{x}) p(\mathbf{d}) dx = \frac{1}{N^2} \sum_{p=1}^{N} \frac{J_p}{N} \sum_{l=1}^{N} \sum_{k=1}^{J_p} N(x_l - x_{pk}, 2\sigma^2 I)$

In practical applications, σ is usually set to the half of the maximum distance between samples as proposed by Torkkola [89].

Appendix D

UCI decoding evaluation performances

Tables D.1 to D.16 show the performance results for the UCI data sets using Gentle Adaboost. The results using Linear SVM are shown in tables from D.17 to D.32.

Table D.1: Dermathology performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	92.04(2.32)	89.37(1.89)	91.04(2.37)	58.84(2.07)	91.82(2.33)	92.04(2.17)
IHD	91.59(2.07)	88.05(1.93)	92.37(1.85)	63.42(1.27)	91.72(2.17)	92.02(2.12)
ED	92.04(2.32)	89.37(1.89)	91.04(2.37)	63.69(1.11)	92.04(2.19)	92.04(2.32)
AED	92.04(2.32)	89.37(1.89)	91.04(2.37)	64.79(0.94)	92.04(2.26)	92.04(2.34)
LLB	91.79(2.39)	95.13(1.11)	94.00(1.76)	54.98(1.79)	90.11(2.32)	91.08(2.22)
ELB	92.07(2.14)	95.13(1.11)	94.00(1.76)	58.78(2.17)	92.04(2.25)	92.04(2.39)
PD	91.32(2.39)	95.11(1.86)	92.62(2.19)	44.49(3.58)	92.04(2.05)	91.75(2.04)
LAP	92.04(2.32)	89.37(1.89)	91.04(2.37)	63.69(1.11)	92.04(2.20)	92.04(2.20)
$\beta - DEN$	92.04(2.32)	89.37(1.89)	91.04(2.37)	63.69(1.11)	92.04(2.04)	92.04(2.11)
$LLWDiscrete$	92.04(2.32)	88.57(1.85)	91.59(2.39)	65.07(0.86)	92.04(2.02)	92.04(2.11)
$LLWContinuous$	91.79(2.39)	95.13(1.11)	93.72(1.99)	45.28(4.14)	91.54(2.13)	91.98(2.28)
$ELWDiscrete$	91.77(2.33)	89.39(1.38)	91.59(2.39)	65.07(0.86)	91.72(2.33)	91.81(2.04)
$ELWContinuous$	92.07(2.04)	95.13(1.11)	94.00(1.76)	43.61(4.24)	92.07(2.38)	92.07(2.25)

Table D.2: Iris performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	92.44(2.26)	93.33(2.18)
IHD	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	92.44(2.26)	93.33(2.18)
ED	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	92.44(2.26)	94.00(2.18)
AED	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	93.33(2.03)	95.33(1.14)
LLB	94.00(2.48)	95.33(1.14)	95.33(1.14)	95.33(1.14)	92.44(2.26)	94.00(2.18)
ELB	94.00(2.48)	95.33(1.14)	95.33(1.14)	95.33(1.14)	92.44(2.26)	94.00(2.18)
PD	94.00(2.48)	93.33(1.14)	95.33(1.14)	95.33(1.14)	92.44(2.26)	94.00(2.18)
LAP	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	93.33(2.27)	95.33(1.14)
$\beta - DEN$	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	93.33(2.18)	95.33(1.14)
$LLWDiscrete$	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	93.33(2.18)	95.33(1.14)
$LLWContinuous$	94.00(2.48)	95.33(1.14)	95.33(1.14)	95.33(1.14)	94.00(2.48)	96.00(1.14)
$ELWDiscrete$	94.00(2.48)	93.33(2.18)	93.33(2.18)	93.33(1.95)	93.33(2.18)	96.00(1.14)
$ELWContinuous$	94.00(2.48)	95.33(1.14)	95.33(1.14)	95.33(1.14)	95.33(1.14)	96.00(1.44)

Table D.3: Ecoli performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	77.87(1.69)	76.45(1.98)	79.46(1.63)	37.32(1.71)	78.01(2.01)	77.46(1.96)
IHD	77.28(1.56)	76.45(1.98)	75.24(1.96)	37.93(1.63)	78.12(2.06)	76.24(1.76)
ED	77.87(1.69)	76.45(1.98)	79.46(1.63)	36.73(2.15)	78.23(1.91)	77.82(2.02)
AED	77.87(1.69)	76.45(1.98)	79.46(1.63)	53.34(1.49)	78.66(1.67)	79.74(1.63)
LLB	81.38(1.37)	80.53(1.56)	81.00(1.19)	28.44(1.93)	78.12(1.58)	77.82(2.02)
ELB	80.53(1.27)	80.53(1.56)	80.39(1.89)	30.22(2.05)	78.92(2.11)	78.02(1.99)
PD	81.47(1.48)	78.44(2.04)	77.73(1.98)	49.33(1.76)	78.42(1.53)	76.42(2.01)
LAP	77.87(1.69)	76.45(1.98)	79.46(1.63)	37.03(1.89)	78.77(1.88)	79.46(1.63)
$\beta - DEN$	77.87(1.69)	76.45(1.98)	79.46(1.63)	35.78(1.21)	78.93(1.85)	79.46(1.63)
$LLWDiscrete$	78.18(1.65)	75.57(1.97)	79.74(1.69)	53.96(1.58)	79.04(1.15)	79.74(1.69)
$LLWContinuous$	81.38(1.11)	78.48(1.45)	80.98(1.74)	44.34(2.21)	78.93(1.44)	80.98(1.74)
$ELWDiscrete$	78.76(1.64)	75.57(1.97)	79.74(1.69)	55.15(1.14)	78.93(1.44)	79.74(1.69)
$ELWContinuous$	80.53(0.89)	79.09(1.44)	80.68(1.89)	47.74(2.93)	79.22(1.56)	80.98(1.74)

Table D.4: Wine performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	94.35(0.81)	94.35(0.81)
IHD	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	94.35(0.81)	94.35(0.81)
ED	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	95.49(1.37)	95.49(1.37)
AED	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	96.05(1.18)	96.05(1.18)
LLB	93.79(1.27)	96.05(1.18)	96.05(1.18)	96.05(1.18)	94.35(0.81)	94.35(0.81)
ELB	94.35(0.98)	95.49(1.37)	95.49(1.37)	95.49(1.37)	94.35(0.81)	94.35(0.81)
PD	95.46(1.11)	95.49(1.37)	95.49(1.37)	95.49(1.37)	94.35(0.81)	94.35(0.81)
LAP	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	96.05(1.18)	96.05(1.18)
$\beta - DEN$	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	96.05(1.18)	96.05(1.18)
$LLWDiscrete$	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	96.05(1.18)	96.05(1.18)
$LLWContinuous$	93.79(1.27)	96.05(1.18)	96.05(1.18)	96.05(1.18)	96.05(1.18)	96.05(1.18)
$ELWDiscrete$	94.35(0.81)	95.49(1.37)	94.93(1.27)	94.35(1.84)	96.05(1.18)	96.05(1.18)
$ELWContinuous$	94.35(0.98)	96.05(1.18)	96.05(1.18)	96.05(1.18)	96.05(1.18)	96.05(1.18)

Table D.5: Glass performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	66.69(3.16)	57.51(3.78)	56.00(4.95)	59.22(3.72)	64.56(2.15)	63.53(2.22)
IHD	67.14(3.08)	53.73(3.77)	45.18(5.32)	56.97(4.04)	64.35(2.56)	65.53(2.84)
ED	66.69(3.16)	57.51(3.78)	56.00(4.95)	59.22(3.72)	64.50(3.25)	66.50(2.78)
AED	66.69(3.16)	57.51(3.78)	56.00(4.95)	58.80(3.64)	65.55(2.87)	66.50(2.78)
LLB	49.31(1.25)	63.68(4.05)	57.87(4.28)	48.64(3.61)	62.64(3.16)	64.01(2.84)
ELB	55.57(3.11)	64.59(4.14)	58.35(3.43)	59.83(3.98)	64.01(2.84)	64.01(2.84)
PD	66.21(2.62)	62.16(2.86)	57.74(4.71)	61.40(3.32)	63.26(2.93)	64.35(2.72)
LAP	66.69(3.16)	57.51(3.78)	56.00(4.95)	59.22(3.72)	66.50(2.78)	66.50(2.78)
$\beta - DEN$	66.69(3.16)	57.51(3.78)	56.00(4.95)	59.22(3.72)	66.50(2.78)	66.50(2.78)
$LLWDiscrete$	67.16(3.12)	60.26(3.67)	52.75(4.01)	59.22(3.72)	66.50(2.78)	66.69(3.16)
$LLWContinuous$	49.31(2.25)	64.01(3.79)	57.85(3.98)	53.04(2.85)	66.50(2.78)	66.69(3.16)
$ELWDiscrete$	67.62(3.02)	55.98(5.08)	53.23(4.28)	59.22(3.72)	66.50(2.78)	66.69(3.16)
$ELWContinuous$	56.03(3.19)	65.01(3.74)	57.85(3.98)	60.50(3.67)	66.69(3.16)	66.69(3.16)

Table D.6: Thyroid performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
IHD	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
ED	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
AED	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
LLB	91.17(3.08)	92.10(2.72)	92.10(2.72)	92.10(2.72)	91.17(3.08)	91.17(3.08)
ELB	91.17(3.08)	92.10(2.72)	92.10(2.72)	92.10(2.72)	91.17(3.08)	91.17(3.08)
PD	92.10(2.63)	91.19(2.67)	91.19(2.67)	91.19(2.67)	91.17(3.08)	91.17(3.08)
LAP	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
$\beta - DEN$	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
$LLWDiscrete$	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
$LLWContinuous$	91.17(3.08)	92.10(2.72)	92.10(2.72)	92.10(2.72)	91.17(3.08)	91.17(3.08)
$ELWDiscrete$	92.10(3.13)	90.71(2.62)	90.71(2.62)	90.71(2.62)	92.10(3.13)	92.10(3.13)
$ELWContinuous$	91.17(3.08)	92.10(2.72)	92.10(2.72)	92.10(2.72)	91.17(3.08)	91.17(3.08)

Table D.7: Vowel performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	59.19(2.83)	42.42(2.28)	27.47(2.07)	38.28(2.08)	60.36(2.67)	61.30(2.67)
IHD	57.98(2.59)	43.33(2.29)	24.14(1.86)	27.98(1.26)	61.02(2.76)	60.82(2.57)
ED	59.19(2.83)	42.42(2.28)	27.47(2.07)	44.14(2.27)	62.83(2.62)	62.56(2.87)
AED	59.19(2.83)	42.42(2.28)	27.47(2.07)	43.03(2.56)	63.25(2.92)	64.36(2.62)
LLB	52.32(3.38)	47.47(2.40)	32.32(2.28)	36.26(2.38)	54.36(3.12)	55.63(3.06)
ELB	55.45(3.42)	47.37(2.43)	33.23(2.39)	39.60(2.48)	55.45(3.19)	56.48(3.12)
PD	58.48(3.02)	45.05(2.27)	31.52(2.29)	43.94(2.04)	56.36(3.22)	55.75(3.21)
LAP	59.19(2.83)	42.42(2.28)	27.47(2.07)	44.34(2.19)	64.91(2.68)	65.36(2.17)
$\beta - DEN$	59.19(2.83)	42.42(2.28)	27.47(2.07)	44.34(2.19)	65.12(2.62)	65.36(2.17)
$LLWDiscrete$	59.09(2.84)	45.66(3.41)	29.70(2.09)	45.15(2.44)	66.71(2.63)	66.79(2.25)
$LLWContinuous$	52.42(3.42)	48.89(2.53)	33.23(2.48)	40.51(2.24)	67.13(3.21)	69.53(3.11)
$ELWDiscrete$	59.70(2.82)	45.66(3.41)	29.70(2.09)	44.95(2.58)	66.93(2.53)	68.45(2.59)
$ELWContinuous$	55.25(3.31)	48.48(2.53)	33.13(2.44)	43.13(1.99)	69.87(3.06)	71.77(3.02)

Table D.8: Balance performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	78.97(5.02)	47.16(5.49)	50.49(6.19)	42.54(2.08)	78.97(5.02)	78.97(5.02)
IHD	78.97(5.02)	47.16(5.49)	50.49(6.19)	42.54(2.08)	78.97(5.02)	78.97(5.02)
ED	78.97(5.02)	47.16(5.49)	50.49(6.19)	80.15(4.01)	78.97(5.02)	78.97(5.02)
AED	78.97(5.02)	47.16(5.49)	50.49(6.19)	80.15(4.01)	78.97(5.02)	78.97(5.02)
LLB	75.93(4.72)	71.30(5.79)	73.08(6.97)	48.10(2.76)	75.93(4.72)	75.93(4.72)
ELB	77.86(4.53)	72.11(7.96)	72.10(7.98)	63.96(4.86)	77.86(4.53)	77.86(4.53)
PD	82.22(4.19)	79.89(7.76)	80.05(7.79)	78.54(3.79)	82.22(4.19)	82.22(4.19)
LAP	78.97(5.02)	47.16(5.49)	50.49(6.19)	80.15(4.01)	78.97(5.02)	78.97(5.02)
$\beta - DEN$	78.97(5.02)	47.16(5.49)	50.49(6.19)	80.15(4.01)	78.97(5.02)	78.97(5.02)
$LLWDiscrete$	78.34(4.19)	76.52(7.98)	80.94(8.24)	78.84(5.10)	78.34(4.19)	78.34(4.19)
$LLWContinuous$	76.09(4.67)	74.87(7.77)	75.67(7.75)	71.02(5.01)	76.09(4.67)	76.09(4.67)
$ELWDiscrete$	78.34(4.19)	76.52(7.98)	80.94(8.24)	78.84(5.51)	78.34(4.19)	78.34(4.19)
$ELWContinuous$	77.55(4.46)	74.87(7.77)	75.67(7.75)	73.56(5.24)	77.55(4.46)	77.55(4.46)

Table D.9: Yeast performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	49.57(1.38)	45.87(1.12)	46.84(1.34)	44.30(2.15)	48.77(1.32)	49.64(1.38)
IHD	49.30(1.58)	43.32(1.09)	22.42(1.58)	40.22(1.18)	48.93(1.51)	49.94(1.62)
ED	49.57(1.38)	45.87(1.12)	46.84(1.34)	43.50(2.08)	50.32(1.44)	50.88(2.16)
AED	49.57(1.38)	45.87(1.12)	46.84(1.34)	39.35(0.81)	51.77(1.27)	51.98(1.29)
LLB	50.74(1.47)	46.54(1.43)	48.11(1.24)	46.16(1.46)	50.74(1.38)	49.52(1.33)
ELB	51.81(1.93)	46.68(1.38)	47.91(1.21)	45.53(1.54)	50.74(1.38)	50.89(1.57)
PD	49.66(1.34)	45.95(2.41)	41.57(1.25)	34.55(0.81)	48.47(1.52)	49.85(1.58)
LAP	49.57(1.38)	45.87(1.12)	46.84(1.34)	43.57(1.96)	51.77(1.35)	52.04(1.38)
$\beta - DEN$	49.57(1.38)	45.87(1.12)	46.84(1.34)	43.57(1.96)	51.79(1.37)	52.04(1.55)
$LLWDiscrete$	49.16(1.47)	41.38(1.21)	46.93(1.81)	39.54(1.37)	51.84(1.29)	52.04(1.27)
$LLWContinuous$	49.46(1.24)	47.96(1.01)	45.29(1.31)	40.12(2.02)	51.76(1.51)	51.88(1.57)
$ELWDiscrete$	49.16(1.47)	41.38(1.21)	46.86(1.81)	40.20(1.63)	52.04(1.37)	52.04(1.45)
$ELWContinuous$	51.41(1.73)	49.05(1.31)	45.42(1.35)	40.32(1.81)	51.84(1.19)	52.17(1.86)

Table D.10: Satimage performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	83.25(2.23)	79.37(2.03)	79.91(2.45)	79.56(1.99)	82.05(1.98)	82.15(2.28)
IHD	83.19(2.17)	75.28(1.56)	76.57(2.24)	76.61(1.88)	81.98(2.05)	82.04(2.27)
ED	83.25(2.23)	79.37(2.03)	79.91(2.45)	81.49(1.87)	82.32(2.02)	83.06(2.33)
AED	83.25(2.23)	79.37(2.03)	79.91(2.45)	63.96(1.48)	82.80(2.36)	84.06(2.13)
LLB	83.51(1.71)	83.56(2.03)	84.09(1.91)	76.60(2.56)	81.83(1.79)	83.10(1.19)
ELB	83.23(1.71)	83.57(1.99)	84.24(1.86)	78.71(2.61)	81.88(1.93)	83.43(2.01)
PD	83.31(2.07)	83.15(1.77)	83.90(1.82)	63.75(2.14)	82.14(2.11)	83.23(2.22)
LAP	83.25(2.23)	79.37(2.03)	79.91(2.45)	81.49(1.87)	83.07(2.19)	84.15(2.22)
$\beta - DEN$	83.25(2.23)	79.37(2.03)	79.91(2.45)	81.49(1.87)	83.11(2.28)	84.15(2.09)
$LLWDiscrete$	83.06(2.19)	80.96(2.02)	80.70(2.37)	67.33(2.54)	83.16(2.49)	84.88(2.36)
$LLWContinuous$	83.22(1.76)	83.26(2.07)	83.84(1.97)	63.80(3.46)	83.87(1.88)	85.25(1.65)
$ELWDiscrete$	83.11(2.14)	80.74(2.21)	80.70(2.37)	65.64(1.56)	83.17(2.23)	85.03(2.09)
$ELWContinuous$	84.13(1.68)	83.31(2.01)	83.82(2.00)	70.59(3.04)	84.07(1.70)	85.37(1.87)

Table D.11: Letter performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	88.16(1.71)	82.54(1.62)	84.32(1.56)	80.31(1.68)	87.35(1.66)	88.31(1.58)
IHD	87.68(1.54)	83.91(1.71)	83.35(1.71)	82.10(1.68)	86.43(1.60)	87.37(1.87)
ED	88.96(1.56)	84.34(1.65)	84.53(1.67)	82.31(1.82)	87.56(1.67)	88.85(1.58)
AED	88.96(1.56)	84.34(1.65)	84.53(1.67)	84.13(1.87)	88.16(1.63)	89.03(1.62)
LLB	88.54(1.62)	85.62(1.74)	83.76(1.68)	82.14(1.76)	86.37(1.80)	87.46(1.74)
ELB	88.76(1.62)	85.89(1.54)	84.51(1.57)	82.80(1.71)	86.77(1.67)	87.72(1.84)
PD	87.65(2.01)	84.37(1.52)	82.74(1.71)	81.21(1.78)	85.62(1.63)	86.37(1.63)
LAP	88.96(1.64)	86.89(1.63)	85.73(1.76)	83.42(1.77)	88.76(1.59)	90.12(1.81)
$\beta - DEN$	88.96(1.64)	87.12(1.60)	88.26(1.50)	83.82(1.51)	89.01(1.54)	90.32(1.58)
$LLWDiscrete$	88.96(1.86)	87.28(1.63)	87.85(1.50)	84.82(1.44)	89.43(1.58)	91.09(1.63)
$LLWContinuous$	89.91(1.44)	87.42(1.56)	89.47(1.81)	85.55(1.55)	89.35(1.77)	90.86(1.69)
$ELWDiscrete$	90.61(1.55)	87.64(1.63)	88.59(1.57)	86.28(1.56)	90.10(1.57)	91.12(1.72)
$ELWContinuous$	90.77(1.60)	88.83(1.55)	90.70(1.52)	88.05(1.66)	91.74(1.65)	91.92(1.58)

Table D.12: Pendigits performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	96.72(0.92)	91.08(1.01)	90.34(0.96)	83.92(1.11)	91.05(0.98)	96.32(1.02)
IHD	97.02(1.09)	91.76(1.01)	92.10(1.23)	81.84(1.00)	91.24(1.13)	96.07(0.89)
ED	97.34(1.01)	92.72(1.07)	93.20(1.13)	85.87(0.96)	92.31(1.12)	96.87(1.11)
AED	97.34(1.01)	92.72(1.07)	93.20(1.13)	88.98(1.07)	92.84(1.26)	97.06(1.24)
LLB	96.78(0.91)	91.87(0.93)	91.36(1.02)	83.67(0.94)	92.13(0.89)	95.73(1.00)
ELB	96.87(0.71)	91.95(0.75)	92.37(0.77)	85.72(0.73)	93.24(0.82)	96.72(0.81)
PD	96.98(1.09)	90.66(1.39)	91.87(1.04)	84.82(1.14)	92.18(1.05)	96.01(1.13)
LAP	97.34(0.91)	93.02(0.76)	94.72(1.00)	92.73(1.00)	94.37(0.98)	97.16(0.97)
$\beta - DEN$	97.34(0.91)	93.02(0.89)	94.72(0.91)	93.26(1.06)	94.80(0.80)	97.19(0.97)
$LLWDiscrete$	97.88(0.87)	93.21(0.85)	94.88(0.87)	93.26(0.89)	95.02(0.89)	97.54(0.92)
$LLWContinuous$	97.98(0.86)	93.24(0.82)	95.32(0.86)	94.02(0.98)	95.82(0.88)	97.64(0.92)
$ELWDiscrete$	97.96(0.69)	93.19(0.55)	95.01(0.69)	93.26(0.75)	95.35(0.70)	97.62(0.75)
$ELWContinuous$	98.01(1.01)	93.98(1.06)	95.54(1.01)	94.78(1.06)	96.25(0.95)	97.84(0.97)

Table D.13: Segmentation performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	96.10(0.79)	91.82(1.26)	93.03(0.82)	92.16(1.01)	93.13(1.31)	95.09(1.24)
IHD	96.23(0.76)	92.03(1.21)	93.46(0.95)	92.21(0.91)	93.67(1.02)	95.93(1.23)
ED	96.10(0.79)	91.82(1.26)	93.03(0.82)	92.16(1.01)	93.88(1.05)	96.13(1.38)
AED	96.10(0.79)	91.82(1.26)	93.03(0.82)	89.18(1.27)	94.03(0.95)	96.44(1.24)
LLB	93.98(0.92)	95.54(0.78)	94.50(0.82)	71.13(1.61)	93.54(0.97)	94.04(1.03)
ELB	95.80(0.80)	95.58(0.76)	94.55(0.78)	86.32(0.63)	93.58(0.88)	93.92(0.88)
PD	96.06(0.87)	95.24(0.86)	94.20(0.65)	92.07(1.03)	93.25(0.69)	94.00(0.89)
LAP	96.10(0.79)	91.82(1.26)	93.03(0.82)	92.16(1.01)	94.33(0.80)	96.44(1.06)
$\beta - DEN$	96.10(0.79)	91.82(1.26)	93.03(0.82)	92.16(1.01)	94.35(0.70)	96.44(1.17)
$LLWDiscrete$	96.15(0.81)	93.25(1.31)	93.98(0.92)	92.73(1.06)	94.58(1.01)	96.48(1.38)
$LLWContinuous$	93.81(0.95)	95.58(0.70)	95.19(0.89)	86.58(0.80)	95.09(0.93)	96.76(0.90)
$ELWDiscrete$	96.15(0.81)	93.25(1.31)	93.98(0.92)	92.86(0.96)	94.65(0.81)	96.65(1.30)
$ELWContinuous$	95.84(0.80)	95.54(0.74)	95.19(0.89)	92.51(0.89)	95.97(0.90)	97.01(0.78)

Table D.14: OptDigits performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	96.50(0.71)	86.58(2.71)	80.57(2.03)	75.36(1.80)	87.99(2.56)	95.05(1.04)
IHD	96.85(1.10)	85.53(2.23)	66.53(1.96)	71.64(1.69)	87.78(2.67)	95.64(1.13)
ED	96.50(0.71)	86.58(2.71)	80.57(2.03)	79.63(2.28)	88.66(2.90)	96.58(1.32)
AED	96.50(0.71)	86.58(2.71)	80.57(2.03)	74.04(2.33)	89.07(2.80)	96.03(1.25)
LLB	93.11(1.42)	89.61(1.81)	85.91(1.56)	78.70(2.62)	86.73(2.08)	94.03(1.54)
ELB	93.95(0.54)	90.67(1.80)	85.10(1.46)	79.57(2.54)	87.83(2.21)	94.83(1.11)
PD	96.26(0.35)	91.46(1.77)	85.39(1.67)	78.38(2.48)	86.38(1.98)	93.72(0.98)
LAP	96.50(0.71)	86.58(2.71)	83.57(2.03)	79.50(2.36)	89.07(2.80)	96.60(1.03)
$\beta - DEN$	96.50(0.71)	86.58(2.71)	83.57(2.03)	79.63(2.28)	89.07(2.80)	96.60(1.03)
$LLWDiscrete$	96.80(0.41)	88.49(3.18)	84.43(2.10)	76.96(2.17)	90.27(2.69)	96.90(1.12)
$LLWContinuous$	93.11(1.42)	92.83(1.64)	86.89(1.64)	79.41(2.58)	91.09(1.87)	96.83(1.32)
$ELWDiscrete$	96.80(0.41)	88.58(3.02)	84.43(2.10)	77.31(2.18)	91.02(2.63)	99.82(1.04)
$ELWContinuous$	93.95(0.54)	92.81(1.69)	86.85(1.66)	80.92(2.05)	91.20(3.02)	97.05(1.01)

Table D.15: Vehicle performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	72.34(3.34)	66.32(3.18)	64.06(2.57)	65.85(2.37)	69.77(3.37)	72.34(3.34)
IHD	72.34(3.34)	67.50(2.68)	65.60(2.05)	70.34(2.47)	69.77(3.37)	72.34(3.34)
ED	72.34(3.34)	66.32(3.18)	64.06(2.57)	70.22(2.36)	69.77(3.37)	72.34(3.34)
AED	72.34(3.34)	66.32(3.18)	64.06(2.57)	71.06(3.69)	69.77(3.37)	72.34(3.34)
LLB	72.35(3.25)	71.99(3.91)	71.75(3.61)	65.36(2.19)	69.77(3.37)	71.08(3.43)
ELB	72.58(3.17)	72.10(3.74)	72.10(3.71)	66.90(2.22)	69.77(3.37)	72.80(3.14)
PD	71.99(3.13)	72.70(2.84)	72.58(2.95)	71.63(2.75)	69.77(3.37)	72.05(3.11)
LAP	72.34(3.34)	66.32(3.18)	64.06(2.57)	70.22(2.36)	69.77(3.37)	72.34(3.34)
$\beta - DEN$	72.34(3.34)	66.32(3.18)	64.06(2.57)	70.22(2.36)	69.77(3.37)	72.34(3.34)
$LLWDiscrete$	72.69(3.62)	70.58(3.65)	71.40(4.16)	70.22(2.36)	69.77(3.37)	72.70(3.62)
$LLWContinuous$	72.23(3.07)	72.45(3.53)	72.46(4.46)	69.14(1.99)	69.77(3.37)	72.70(3.62)
$ELWDiscrete$	72.69(3.62)	70.58(3.65)	71.40(4.16)	70.22(2.36)	69.77(3.37)	72.70(3.62)
$ELWContinuous$	72.35(3.17)	72.57(3.74)	72.33(4.42)	69.97(1.94)	69.77(3.37)	73.15(3.16)

Table D.16: Shuttle performance using Gentle Adaboost.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	99.84(0.07)	99.83(0.07)	99.74(0.10)	85.02(0.07)	99.84(0.07)	99.84(0.07)
IHD	99.68(0.28)	99.83(0.07)	99.72(0.07)	84.97(0.07)	99.68(0.28)	99.68(0.28)
ED	99.84(0.07)	99.83(0.07)	99.74(0.10)	99.87(0.06)	99.84(0.07)	99.84(0.07)
AED	99.84(0.07)	99.83(0.07)	99.74(0.10)	99.82(0.07)	99.84(0.07)	99.84(0.07)
LLB	91.43(0.42)	99.88(0.07)	99.84(0.07)	85.45(0.28)	93.42(0.35)	91.43(0.42)
ELB	98.92(1.03)	99.88(0.07)	99.85(0.07)	85.94(0.47)	98.76(1.24)	98.92(1.03)
PD	99.15(0.60)	95.38(1.42)	97.91(3.13)	94.45(0.88)	98.73(0.82)	99.15(0.07)
LAP	99.84(0.07)	99.83(0.07)	99.74(0.10)	99.87(0.07)	99.84(0.07)	99.84(0.07)
$\beta - DEN$	99.84(0.07)	99.83(0.07)	99.74(0.10)	99.87(0.07)	99.84(0.07)	99.84(0.07)
$LLWDiscrete$	99.85(0.07)	99.84(0.07)	99.78(0.07)	99.88(0.07)	99.85(0.07)	99.85(0.07)
$LLWContinuous$	94.31(2.14)	99.88(0.07)	99.85(0.07)	93.67(2.23)	99.85(0.07)	99.85(0.07)
$ELWDiscrete$	99.86(0.07)	99.84(0.07)	99.78(0.07)	99.88(0.07)	99.86(0.07)	99.86(0.07)
$ELWContinuous$	99.52(0.20)	99.88(0.07)	99.85(0.07)	99.01(0.07)	99.86(0.07)	99.86(0.07)

Table D.17: Dermathology performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	95.59(0.74)	94.54(1.04)	80.86(1.26)	37.43(0.57)	94.90(0.63)	94.88(1.03)
IHD	95.03(1.07)	95.11(1.27)	30.60(1.30)	38.27(0.77)	93.82(1.16)	93.51(1.07)
ED	95.59(0.74)	94.54(1.04)	80.86(1.26)	44.02(2.49)	95.07(1.02)	95.52(0.94)
AED	95.59(0.74)	94.54(1.04)	80.86(1.26)	86.81(1.70)	96.10(1.04)	95.59(0.75)
LLB	84.82(1.52)	96.12(0.93)	81.41(1.17)	45.60(3.26)	91.84(1.57)	89.26(1.04)
ELB	94.22(1.12)	96.12(0.93)	81.41(1.17)	62.33(2.70)	92.32(1.43)	93.56(1.33)
PD	94.47(1.30)	93.39(1.30)	80.88(1.09)	85.38(3.02)	91.41(1.07)	91.73(1.24)
LAP	95.59(0.74)	94.54(1.04)	80.86(1.26)	44.02(2.49)	96.10(0.94)	95.59(1.00)
$\beta - DEN$	95.59(0.74)	94.54(1.04)	80.86(1.26)	71.91(2.43)	96.10(0.94)	96.10(0.83)
$LLWDiscrete$	95.59(0.74)	95.10(1.01)	80.86(1.26)	87.09(1.50)	96.20(0.88)	96.31(0.87)
$LLWContinuous$	84.82(1.52)	96.12(0.93)	81.41(1.17)	76.68(3.03)	96.03(1.42)	96.20(1.42)
$ELWDiscrete$	95.59(0.74)	95.38(0.86)	80.86(1.26)	87.09(1.50)	96.26(1.03)	96.31(1.07)
$ELWContinuous$	94.22(1.12)	96.12(0.93)	81.41(1.17)	83.53(3.11)	96.31(1.07)	96.40(1.60)

Table D.18: Iris performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	97.33(1.07)	72.00(2.54)	72.00(2.54)	66.67(1.00)	97.33(1.07)	97.33(1.07)
IHD	97.33(1.07)	72.00(2.54)	72.00(2.54)	66.67(1.00)	97.33(1.07)	97.33(1.07)
ED	97.33(1.07)	72.00(2.54)	72.00(2.54)	97.33(1.07)	97.33(1.07)	97.33(1.07)
AED	97.33(1.07)	72.00(2.54)	72.00(2.54)	97.33(1.07)	97.33(1.07)	97.33(1.07)
LLB	58.00(1.07)	92.67(2.05)	92.67(2.05)	66.67(1.00)	58.00(1.70)	58.00(1.70)
ELB	97.33(1.07)	92.67(2.05)	92.67(2.05)	92.00(1.90)	97.33(1.07)	97.33(1.07)
PD	97.33(1.07)	82.67(2.22)	82.67(2.22)	77.33(1.99)	97.33(1.07)	97.33(1.07)
LAP	97.33(1.07)	72.00(2.54)	72.00(2.54)	97.33(1.07)	97.33(1.07)	97.33(1.07)
$\beta - DEN$	97.33(1.07)	72.00(2.54)	72.00(2.54)	97.33(1.07)	97.33(1.07)	97.33(1.07)
$LLWDiscrete$	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)
$LLWContinuous$	58.00(1.70)	94.00(2.05)	94.00(2.05)	78.00(2.39)	58.00(1.70)	58.00(1.70)
$ELWDiscrete$	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)	97.33(1.07)
$ELWContinuous$	97.33(1.07)	93.33(2.18)	93.33(2.18)	93.33(1.69)	97.33(1.07)	97.33(1.07)

Table D.19: Ecoli performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	77.02(3.33)	61.60(3.37)	68.12(3.36)	68.42(2.26)	77.23(2.64)	77.80(3.06)
IHD	77.02(3.33)	60.12(3.37)	72.15(2.86)	26.92(2.57)	76.01(2.36)	75.36(3.52)
ED	77.02(3.33)	61.60(3.37)	68.12(3.36)	72.63(3.16)	78.04(2.91)	78.55(2.51)
AED	77.02(3.33)	61.60(3.37)	68.12(3.36)	72.31(3.06)	78.90(3.00)	79.98(3.08)
LLB	75.56(2.76)	78.81(2.57)	76.42(3.45)	69.98(2.69)	74.53(2.94)	76.59(2.73)
ELB	76.13(3.07)	78.81(2.57)	76.42(3.45)	74.34(3.14)	75.35(3.16)	77.48(3.21)
PD	77.92(2.27)	82.44(1.91)	78.53(1.86)	27.50(2.90)	73.42(2.44)	76.98(2.53)
LAP	77.02(3.33)	61.60(3.37)	68.12(3.36)	72.63(3.16)	79.02(2.86)	80.24(2.95)
$\beta - DEN$	77.02(3.33)	61.60(3.37)	68.12(3.36)	72.63(3.16)	79.11(2.90)	80.33(2.51)
$LLWDiscrete$	78.64(2.49)	68.81(3.42)	71.52(2.79)	68.38(3.80)	79.22(2.45)	80.41(2.60)
$LLWContinuous$	80.45(2.39)	70.86(3.28)	70.44(3.50)	41.32(3.33)	79.30(2.76)	81.82(2.85)
$ELWDiscrete$	79.37(2.51)	66.35(3.75)	67.79(3.44)	61.35(4.88)	79.35(3.06)	80.70(2.93)
$ELWContinuous$	81.30(2.62)	73.05(3.21)	70.36(4.88)	41.10(4.23)	80.90(2.71)	81.94(2.50)

Table D.20: Wine performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	94.75(1.56)	94.75(1.56)
IHD	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	94.75(1.56)	94.75(1.56)
ED	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	94.75(1.56)	94.75(1.56)
AED	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	95.55(1.49)	95.55(1.49)
LLB	90.97(2.10)	95.55(1.34)	95.55(1.34)	95.55(1.34)	94.75(1.56)	94.75(1.56)
ELB	92.64(1.70)	95.55(1.34)	95.55(1.34)	95.55(1.34)	94.75(1.56)	94.75(1.56)
PD	93.82(1.27)	94.96(0.95)	94.96(0.95)	94.96(0.95)	94.75(1.56)	94.75(1.56)
LAP	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	95.55(1.49)	95.55(1.49)
$\beta - DEN$	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	95.55(1.49)	95.55(1.49)
$LLWDiscrete$	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	95.55(1.49)	95.55(1.49)
$LLWContinuous$	90.97(2.10)	95.55(1.34)	95.55(1.34)	95.55(1.34)	95.55(1.49)	95.55(1.49)
$ELWDiscrete$	93.78(1.76)	93.23(1.63)	93.23(1.63)	93.23(1.63)	95.55(1.49)	95.55(1.49)
$ELWContinuous$	92.64(1.70)	95.55(1.34)	95.55(1.34)	95.55(1.34)	95.55(1.49)	95.55(1.49)

Table D.21: Glass performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	55.75(3.60)	48.02(2.30)	40.04(3.08)	43.83(2.45)	52.71(3.02)	55.18(3.16)
IHD	55.77(3.72)	36.35(3.86)	32.79(1.54)	46.07(2.94)	51.09(3.47)	54.52(3.18)
ED	55.75(3.60)	48.02(2.30)	40.04(3.08)	44.60(3.39)	52.99(2.51)	55.90(2.51)
AED	55.75(3.60)	48.02(2.30)	40.04(3.08)	45.72(2.38)	54.40(2.52)	57.47(2.69)
LLB	45.85(2.73)	51.48(3.36)	34.91(3.24)	42.87(2.53)	51.21(2.56)	56.46(3.02)
ELB	57.54(3.03)	51.48(3.36)	34.91(3.24)	45.67(2.95)	52.94(2.86)	57.66(3.12)
PD	57.04(3.59)	51.84(4.58)	33.40(5.46)	45.29(2.29)	51.93(3.09)	56.40(3.19)
LAP	55.75(3.60)	48.02(2.30)	40.04(3.08)	45.20(2.87)	55.93(3.06)	57.73(3.14)
$\beta - DEN$	55.75(3.60)	48.02(2.30)	40.04(3.08)	44.60(3.39)	56.04(2.68)	57.92(3.46)
$LLWDiscrete$	50.57(3.26)	41.36(3.64)	37.79(2.89)	44.38(2.16)	54.36(3.16)	58.03(3.07)
$LLWContinuous$	57.84(3.89)	40.43(3.66)	41.45(3.86)	45.77(3.09)	56.80(2.94)	59.04(3.14)
$ELWDiscrete$	58.14(4.47)	41.36(3.64)	37.38(3.76)	42.78(3.15)	57.89(2.86)	59.13(3.19)
$ELWContinuous$	58.65(2.66)	42.87(3.41)	41.50(3.47)	45.81(3.31)	58.15(3.32)	59.30(3.16)

Table D.22: Thyroid performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	94.39(2.15)	91.65(2.67)	91.65(2.67)	81.41(0.94)	94.39(2.15)	94.39(2.15)
IHD	94.39(2.15)	91.65(2.67)	91.65(2.67)	81.41(0.94)	94.39(2.15)	94.39(2.15)
ED	94.39(2.15)	91.65(2.67)	91.65(2.67)	94.39(2.15)	94.39(2.15)	94.39(2.15)
AED	94.39(2.15)	91.65(2.67)	91.65(2.67)	94.39(2.15)	94.39(2.15)	94.39(2.15)
LLB	77.23(1.00)	93.46(2.57)	93.46(2.57)	81.88(0.76)	77.23(1.00)	79.86(1.07)
ELB	92.87(2.22)	93.46(2.57)	93.46(2.57)	94.80(1.59)	92.87(2.22)	92.97(2.32)
PD	93.48(2.49)	85.58(1.98)	85.58(1.98)	89.76(2.31)	93.48(2.49)	93.41(2.46)
LAP	94.39(2.15)	91.65(2.67)	91.65(2.67)	94.39(2.15)	94.39(2.15)	94.39(2.15)
$\beta - DEN$	94.39(2.15)	91.65(2.67)	91.65(2.67)	94.39(2.15)	94.39(2.15)	94.39(2.15)
$LLWDiscrete$	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)
$LLWContinuous$	77.23(1.00)	93.44(2.58)	93.44(2.58)	85.63(1.19)	77.23(1.00)	94.87(2.22)
$ELWDiscrete$	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)	94.39(2.15)
$ELWContinuous$	94.87(2.22)	93.46(2.57)	93.46(2.57)	94.85(2.00)	94.87(2.22)	94.87(2.22)

Table D.23: Vowel performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	64.95(3.71)	26.67(2.11)	28.18(3.16)	34.34(1.86)	65.73(2.62)	65.46(2.67)
IHD	64.95(3.48)	32.63(2.31)	22.83(1.62)	30.81(1.85)	64.33(3.11)	65.12(2.90)
ED	64.95(3.71)	26.67(2.11)	28.18(3.16)	34.14(1.73)	66.78(2.67)	66.90(2.73)
AED	64.95(3.71)	26.67(2.11)	28.18(3.16)	30.10(2.19)	68.26(2.64)	67.79(2.57)
LLB	31.52(2.65)	43.03(3.10)	33.23(2.56)	31.31(2.16)	67.28(2.99)	67.34(2.79)
ELB	64.95(3.19)	43.03(3.10)	32.93(2.72)	29.80(2.08)	67.82(2.89)	67.56(2.70)
PD	63.64(3.43)	40.51(2.90)	32.42(2.33)	24.18(2.52)	65.36(3.08)	66.37(2.69)
LAP	64.95(3.71)	26.67(2.11)	28.18(3.16)	35.25(1.86)	68.36(3.02)	68.40(2.94)
$\beta - DEN$	64.95(3.71)	26.67(2.11)	28.18(3.16)	35.25(1.86)	68.36(3.08)	68.53(3.13)
$LLWDiscrete$	65.96(3.61)	32.83(2.10)	31.11(3.05)	35.76(2.54)	69.38(2.91)	68.47(3.02)
$LLWContinuous$	31.21(2.61)	38.99(3.33)	38.59(2.92)	32.83(2.99)	69.73(2.62)	69.50(2.83)
$ELWDiscrete$	65.96(3.61)	32.83(2.10)	31.11(3.05)	35.96(2.51)	70.82(2.86)	69.88(2.96)
$ELWContinuous$	65.15(3.08)	44.14(2.79)	36.26(2.36)	36.06(2.34)	71.44(3.16)	70.87(3.05)

Table D.24: Balance performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	84.62(4.27)	85.57(4.18)
IHD	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	84.62(4.27)	85.57(4.18)
ED	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	82.83(4.39)	85.57(4.18)
AED	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)
LLB	82.83(4.39)	83.36(3.52)	83.36(3.52)	83.36(3.52)	82.83(4.39)	83.36(3.52)
ELB	85.43(4.32)	83.36(3.52)	83.36(3.52)	83.36(3.52)	85.43(4.32)	83.36(3.52)
PD	83.36(4.09)	82.23(4.01)	82.23(4.01)	82.23(4.01)	83.36(4.09)	82.23(4.01)
LAP	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)
$\beta - DEN$	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)
$LLWDiscrete$	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)
$LLWContinuous$	84.62(4.27)	84.59(4.59)	84.59(4.59)	84.59(4.59)	85.57(4.18)	85.57(4.18)
$ELWDiscrete$	84.62(4.27)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)
$ELWContinuous$	84.94(4.31)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)	85.57(4.18)

Table D.25: Yeast performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	50.79(2.39)	35.52(1.00)	27.82(1.59)	38.00(1.24)	50.23(2.51)	50.35(2.42)
IHD	51.12(2.46)	35.52(1.00)	16.96(1.62)	38.55(1.35)	50.34(2.56)	50.76(2.44)
ED	50.79(2.39)	35.52(1.00)	27.82(1.59)	38.07(1.25)	50.79(2.48)	51.04(2.51)
AED	50.79(2.39)	35.52(1.00)	27.82(1.59)	37.80(1.38)	50.79(2.44)	52.20(2.57)
LLB	50.26(2.31)	50.10(1.13)	26.53(1.53)	37.99(1.64)	50.34(2.45)	50.54(2.62)
ELB	50.38(2.31)	50.10(1.13)	26.67(1.44)	39.08(1.29)	50.63(2.62)	50.98(2.59)
PD	51.17(1.41)	51.64(2.83)	33.65(1.27)	21.76(0.57)	50.73(2.41)	50.30(2.45)
LAP	50.79(2.39)	35.52(1.00)	27.82(1.59)	38.07(1.25)	50.79(2.21)	52.20(2.44)
$\beta - DEN$	50.79(2.39)	35.52(1.00)	27.82(1.59)	38.13(1.31)	50.79(2.28)	52.34(2.46)
$LLWDiscrete$	51.18(0.42)	17.09(3.48)	40.50(1.21)	34.12(1.75)	51.14(2.66)	52.10(2.37)
$LLWContinuous$	52.58(2.08)	50.13(1.10)	45.12(1.77)	21.08(0.95)	52.43(2.68)	52.38(2.62)
$ELWDiscrete$	49.43(2.84)	35.58(1.04)	40.78(1.06)	34.66(2.14)	52.17(2.56)	52.21(2.65)
$ELWContinuous$	51.36(2.61)	48.70(0.92)	44.83(1.77)	22.16(0.61)	52.45(2.60)	52.63(2.45)

Table D.26: Satimage performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	83.36(2.02)	70.71(2.16)	72.01(1.25)	71.87(1.65)	74.25(2.12)	81.92(2.06)
IHD	83.32(2.02)	76.81(2.74)	70.24(1.88)	71.03(1.47)	75.36(2.18)	80.37(2.13)
ED	83.36(2.02)	70.71(2.16)	72.01(1.25)	72.60(1.42)	76.38(2.17)	82.03(2.00)
AED	83.36(2.02)	70.71(2.16)	72.01(1.25)	61.52(1.95)	77.89(1.51)	83.02(1.93)
LLB	62.22(2.79)	77.25(2.29)	77.31(1.44)	67.06(3.89)	70.89(1.79)	72.74(2.01)
ELB	83.88(2.13)	77.25(2.29)	77.42(1.43)	72.50(1.91)	73.74(1.97)	78.73(2.07)
PD	80.90(1.74)	77.97(1.50)	74.83(0.99)	63.45(2.15)	72.15(1.73)	77.83(2.11)
LAP	83.36(2.02)	70.71(2.16)	72.01(1.25)	72.60(1.42)	78.03(1.97)	83.20(1.97)
$\beta - DEN$	83.36(2.02)	70.71(2.16)	72.01(1.25)	72.60(1.42)	79.83(2.08)	83.27(2.05)
$LLWDiscrete$	83.39(2.04)	75.06(2.19)	79.56(2.63)	61.85(1.99)	80.93(1.93)	83.33(2.07)
$LLWContinuous$	60.28(2.14)	67.32(4.34)	61.79(2.44)	40.31(2.88)	81.73(2.62)	83.17(2.09)
$ELWDiscrete$	83.39(2.04)	75.06(2.19)	79.56(2.63)	63.87(3.03)	81.31(2.12)	83.49(2.17)
$ELWContinuous$	83.88(2.12)	78.24(2.52)	80.61(2.56)	66.08(2.18)	82.15(2.05)	84.07(2.00)

Table D.27: Letter performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	85.09(0.88)	36.38(0.76)	65.73(0.86)	63.27(1.01)	80.01(0.84)	85.11(0.97)
IHD	86.01(0.89)	36.93(0.72)	66.71(0.83)	63.74(0.94)	81.17(0.84)	84.93(0.91)
ED	86.11(0.99)	36.38(0.76)	67.28(0.81)	64.83(0.89)	82.09(0.81)	86.25(0.86)
AED	86.11(0.99)	36.38(0.76)	67.28(0.81)	66.27(0.95)	84.21(0.99)	88.71(0.91)
LLB	49.08(1.09)	60.88(0.88)	65.26(0.86)	64.89(1.46)	78.19(0.89)	72.01(0.95)
ELB	85.86(0.91)	60.88(0.88)	66.26(0.86)	65.27(0.95)	79.62(1.00)	79.82(0.80)
PD	68.72(0.91)	51.54(0.73)	65.83(0.94)	64.31(1.06)	77.82(1.01)	74.26(0.95)
LAP	86.22(1.00)	36.38(0.76)	68.73(1.01)	67.10(1.18)	85.15(0.96)	88.89(1.05)
$\beta - DEN$	86.47(0.92)	36.38(0.76)	70.37(0.97)	67.28(0.95)	85.62(0.89)	89.03(0.94)
$LLWDiscrete$	87.64(0.95)	40.85(0.83)	71.26(0.91)	70.21(1.07)	86.19(0.92)	89.12(0.88)
$LLWContinuous$	87.85(1.10)	39.85(0.65)	71.87(0.95)	70.55(0.88)	86.10(0.79)	89.10(0.78)
$ELWDiscrete$	88.61(1.05)	40.81(0.84)	71.66(0.92)	70.37(0.95)	87.21(0.87)	89.27(0.91)
$ELWContinuous$	88.98(0.96)	47.45(0.91)	72.19(0.95)	71.02(0.98)	88.02(1.07)	89.44(0.97)

Table D.28: Pendigits performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	95.73(0.76)	92.37(0.68)	92.18(0.94)	87.82(0.88)	94.28(0.93)	95.87(0.88)
IHD	96.10(0.92)	91.78(0.83)	91.72(1.03)	88.72(0.78)	95.12(0.95)	96.21(0.79)
ED	97.04(0.78)	94.05(0.85)	93.25(0.93)	89.37(0.88)	95.72(0.81)	96.89(0.88)
AED	97.04(0.78)	94.05(0.85)	93.25(0.93)	91.27(0.94)	96.04(0.92)	96.98(0.95)
LLB	95.37(1.09)	93.16(0.92)	91.78(1.01)	90.21(0.97)	93.67(0.98)	94.67(1.01)
ELB	96.21(1.20)	94.28(1.02)	92.16(1.08)	90.21(1.09)	94.26(1.09)	95.37(1.04)
PD	95.63(1.09)	91.72(1.06)	90.36(1.02)	89.71(0.94)	93.62(1.00)	94.20(1.06)
LAP	97.04(0.73)	94.27(0.82)	93.62(0.95)	91.34(0.89)	96.09(0.89)	97.09(0.93)
$\beta - DEN$	97.04(0.82)	94.38(0.87)	93.88(0.92)	91.66(0.92)	96.17(0.92)	97.11(0.94)
$LLWDiscrete$	97.12(0.78)	94.63(0.74)	93.65(0.84)	91.72(0.88)	96.23(0.56)	97.25(0.89)
$LLWContinuous$	97.04(1.44)	95.37(1.03)	94.03(1.11)	92.13(1.46)	96.30(1.05)	97.31(1.08)
$ELWDiscrete$	97.26(1.52)	95.54(1.61)	94.23(1.47)	91.88(1.35)	96.47(1.09)	97.27(1.17)
$ELWContinuous$	97.36(1.27)	95.87(2.30)	95.62(1.29)	92.73(1.37)	96.69(1.07)	97.42(1.09)

Table D.29: Segmentation performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	95.15(0.62)	80.82(1.21)	81.86(0.66)	70.69(1.39)	94.94(1.02)	95.51(0.62)
IHD	95.11(0.59)	80.87(1.20)	80.74(0.59)	62.77(1.58)	95.03(1.05)	94.27(0.62)
ED	95.15(0.62)	80.82(1.21)	81.86(0.66)	83.81(0.67)	95.07(0.99)	95.74(0.62)
AED	95.15(0.62)	80.82(1.21)	81.86(0.66)	80.00(0.59)	95.30(0.89)	96.10(0.59)
LLB	35.11(1.23)	91.69(0.89)	85.15(0.66)	47.53(1.00)	78.02(1.09)	90.37(1.07)
ELB	95.06(0.65)	91.69(0.89)	84.81(0.71)	84.98(0.75)	95.10(0.91)	94.17(0.84)
PD	90.00(0.84)	89.31(0.76)	73.29(0.69)	71.60(0.66)	93.20(0.89)	92.31(0.57)
LAP	95.15(0.62)	80.82(1.21)	81.86(0.66)	83.72(0.73)	95.48(0.93)	96.12(0.65)
$\beta - DEN$	95.15(0.62)	80.82(1.21)	81.86(0.66)	83.72(0.73)	95.76(0.86)	96.34(0.62)
$LLWDiscrete$	95.32(0.66)	90.87(0.89)	85.71(0.69)	83.72(0.73)	96.06(0.95)	96.44(0.65)
$LLWContinuous$	34.59(1.20)	77.06(1.43)	86.71(0.53)	70.00(3.07)	95.83(1.05)	96.10(0.66)
$ELWDiscrete$	95.32(0.66)	90.87(0.89)	85.71(0.69)	83.72(0.73)	96.16(0.94)	96.59(0.77)
$ELWContinuous$	95.02(0.65)	92.21(0.73)	86.58(0.78)	85.19(0.79)	96.43(0.92)	96.98(0.69)

Table D.30: OptDigits performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	96.21(1.18)	89.39(2.38)	82.05(2.87)	71.05(1.95)	91.20(1.89)	96.21(1.08)
IHD	96.23(1.17)	88.91(2.21)	76.67(3.18)	66.47(2.18)	90.26(2.09)	96.21(1.08)
ED	96.21(1.18)	89.39(2.38)	82.05(2.87)	76.78(1.70)	92.06(2.18)	96.21(1.08)
AED	96.21(1.18)	89.39(2.38)	82.05(2.87)	73.86(2.12)	92.30(2.08)	96.21(1.08)
LLB	79.48(3.18)	93.58(1.93)	87.49(2.79)	61.69(2.95)	91.21(2.14)	89.53(2.04)
ELB	96.00(1.17)	93.58(1.93)	87.60(2.82)	76.33(1.24)	92.09(2.15)	96.12(1.13)
PD	95.11(1.16)	94.09(1.57)	87.22(2.56)	71.94(2.00)	91.04(1.87)	95.34(1.24)
LAP	96.21(1.18)	89.39(2.38)	82.05(2.87)	76.92(1.67)	92.70(2.80)	96.77(1.22)
$\beta - DEN$	96.21(1.18)	89.39(2.38)	82.05(2.87)	76.94(1.66)	92.78(2.87)	96.77(1.22)
$LLWDiscrete$	96.23(1.19)	91.19(2.36)	84.59(2.90)	76.99(1.58)	93.09(2.16)	96.77(1.22)
$LLWContinuous$	79.48(3.18)	93.59(1.92)	87.95(2.94)	62.05(1.47)	93.36(2.09)	96.83(2.53)
$ELWDiscrete$	96.21(1.28)	91.23(2.31)	84.57(2.93)	76.85(1.80)	92.09(2.11)	96.77(1.22)
$ELWContinuous$	96.00(1.17)	93.59(1.93)	87.79(2.79)	81.33(2.42)	94.09(2.22)	96.89(1.44)

Table D.31: Vehicle performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	76.12(2.40)	68.67(3.12)	71.27(2.92)	52.01(1.15)	64.00(4.39)	76.12(2.40)
IHD	76.12(2.40)	69.85(2.23)	74.46(2.10)	72.21(1.64)	64.00(4.39)	76.12(2.40)
ED	76.12(2.40)	68.67(3.12)	71.27(2.92)	70.33(2.20)	64.00(4.39)	76.12(2.40)
AED	76.12(2.40)	68.67(3.12)	71.27(2.92)	70.22(2.26)	64.00(4.39)	76.12(2.40)
LLB	71.04(3.20)	78.60(2.40)	74.81(1.84)	67.00(2.66)	65.10(4.31)	71.04(3.20)
ELB	76.47(1.83)	78.72(2.36)	74.93(1.79)	73.04(1.74)	65.70(4.23)	76.47(1.83)
PD	76.00(1.45)	75.06(2.20)	74.23(1.88)	62.06(2.94)	64.37(4.17)	76.00(1.45)
LAP	76.12(2.40)	68.67(3.12)	71.27(2.92)	70.33(2.20)	64.00(4.39)	76.12(2.40)
$\beta - DEN$	76.12(2.40)	68.67(3.12)	71.27(2.92)	70.33(2.20)	64.00(4.39)	76.12(2.40)
$LLWDiscrete$	76.94(2.03)	74.82(2.26)	73.75(2.74)	70.33(2.20)	64.00(4.39)	76.94(2.03)
$LLWContinuous$	71.62(2.63)	78.13(2.21)	74.23(2.78)	70.44(2.59)	66.09(4.80)	76.94(2.03)
$ELWDiscrete$	76.94(2.03)	74.82(2.26)	73.75(2.74)	70.33(2.20)	64.00(4.39)	76.94(2.03)
$ELWContinuous$	76.95(1.76)	79.07(2.23)	74.70(2.03)	72.81(1.44)	66.10(4.71)	76.95(1.76)

Table D.32: Shuttle performance using Linear SVM.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	97.72(0.30)	97.72(0.30)	96.27(0.63)	97.72(0.30)	97.72(0.30)	97.80(0.32)
IHD	97.79(0.28)	97.72(0.30)	95.63(0.44)	97.72(0.30)	97.72(0.30)	97.78(0.29)
ED	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.80(0.32)
AED	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.80(0.32)
LLB	89.43(1.45)	88.72(1.35)	90.94(1.09)	89.20(1.42)	91.43(1.10)	91.82(1.54)
ELB	97.37(0.46)	97.72(0.30)	94.72(0.87)	97.72(0.30)	97.72(0.30)	97.65(0.87)
PD	96.63(0.33)	95.82(0.65)	93.92(0.88)	96.93(0.75)	97.30(1.00)	96.76(0.71)
LAP	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.80(0.32)
$\beta - DEN$	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.80(0.32)
$LLWDiscrete$	97.79(0.27)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.89(0.38)
$LLWContinuous$	97.79(1.44)	97.72(0.30)	97.720.30()	97.72(0.30)	97.72(0.30)	97.98(0.97)
$ELWDiscrete$	97.80(0.28)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.92(0.33)
$ELWContinuous$	97.80(0.44)	97.72(0.30)	97.72(0.30)	97.72(0.30)	97.72(0.30)	98.03(0.77)

Appendix E

Traffic sign categorization performances

Tables E.1 and E.2 show the performance results of the traffic sign experiments using Gentle Adaboost and Linear SVM, respectively.

Table E.1: Gentle Adaboost results for the coding and decoding strategies on the traffic sign data set.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	75.27(3.36)	59.39(2.85)	53.34(1.84)	52.61(2.09)	62.31(2.85)	75.78(2.84)
IHD	75.73(3.60)	60.59(2.97)	47.60(2.27)	48.37(2.17)	63.13(3.02)	74.92(3.28)
ED	75.27(3.36)	59.39(2.85)	53.34(1.84)	54.64(2.38)	63.75(2.85)	76.31(2.79)
AED	75.27(3.36)	59.39(2.85)	53.34(1.84)	53.48(2.22)	67.27(3.33)	78.23(2.86)
LLB	68.25(2.79)	69.12(3.13)	61.04(1.86)	40.42(1.57)	63.00(2.82)	76.23(3.48)
ELB	70.86(2.80)	69.00(3.18)	61.16(1.53)	43.03(1.50)	64.91(2.63)	76.38(3.14)
PD	74.06(3.09)	64.76(4.45)	60.92(1.38)	56.17(2.78)	61.21(2.62)	72.14(3.60)
LAP	75.27(3.36)	59.39(2.85)	53.34(1.84)	54.40(2.39)	68.34(2.93)	79.21(2.64)
$\beta - DEN$	75.27(3.36)	59.39(2.85)	53.34(1.84)	54.64(2.38)	68.34(3.06)	80.10(2.90)
$LLWDiscrete$	75.26(3.31)	63.03(3.63)	55.44(2.38)	52.89(2.20)	69.62(2.91)	80.33(2.97)
$LLWContinuous$	68.38(2.75)	69.83(3.33)	62.11(1.69)	45.07(1.77)	70.43(2.84)	81.03(3.46)
$ELWDiscrete$	75.39(3.16)	63.50(3.38)	55.32(2.37)	53.95(2.08)	70.49(2.95)	80.37(3.32)
$ELWContinuous$	75.86(2.80)	70.06(3.19)	62.23(1.67)	49.20(1.97)	71.88(2.72)	81.26(3.05)

Table E.2: Linear SVM results for the coding and decoding strategies on the traffic sign data set.

	one-versus-one	one-versus-all	dense	sparse	decoc	ecoc-one
HD	83.45(3.08)	67.12(2.54)	64.26(2.62)	66.99(2.41)	75.98(2.91)	82.81(2.87)
IHD	83.92(3.14)	68.30(2.82)	56.12(2.63)	61.16(2.43)	76.21(3.10)	83.01(3.08)
ED	83.45(3.08)	67.12(2.54)	64.26(2.62)	70.53(2.72)	78.21(3.00)	83.33(2.82)
AED	83.45(3.08)	67.12(2.54)	64.26(2.62)	67.33(2.45)	79.92(3.03)	84.26(3.23)
LLB	69.24(2.83)	77.76(2.51)	69.47(2.21)	65.83(2.66)	77.28(2.92)	74.39(3.38)
ELB	77.29(3.56)	77.76(2.51)	69.12(2.16)	69.25(2.90)	77.92(3.38)	75.63(3.77)
PD	78.60(3.72)	68.74(5.02)	70.07(2.24)	65.08(2.17)	76.88(3.24)	76.52(3.94)
LAP	83.45(3.08)	67.12(2.54)	64.26(2.62)	70.06(2.76)	80.33(3.00)	82.91(3.50)
$\beta - DEN$	83.45(3.08)	67.12(2.54)	64.26(2.62)	70.06(2.80)	81.11(3.16)	84.26(3.40)
$LLWDiscrete$	83.45(3.08)	67.12(2.54)	64.26(2.62)	70.89(2.79)	81.03(3.08)	85.36(3.05)
$LLWContinuous$	69.24(2.83)	77.76(2.51)	69.47(2.21)	66.99(2.27)	80.19(2.76)	85.84(3.11)
$ELWDiscrete$	83.10(3.24)	68.79(2.56)	64.61(2.43)	70.29(2.82)	81.14(2.90)	86.82(3.08)
$ELWContinuous$	83.65(3.56)	77.76(2.51)	69.12(2.16)	71.84(2.86)	81.12(3.19)	87.87(3.31)

Appendix F

UCI Machine Learning Repository

The UCI Machine Learning is a repository of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms [8]. The repository contains 111 databases and domain theories. The construction of this repository is an on-going process. The majority of the entries in the repository were contributed by machine learning researchers outside of UCI.

In particular, we selected a set of 16 multi-class data sets from the UCI repository to test de multi-class methodology presented in this thesis. The details of each of the data sets used in the experimental results of the previous sections are shown in table F.1.

Table F.1: UCI repository data sets characteristics.

Problem	#Train	#Attributes	#Classes	Data types	Attribute types	Year
Dermathology	366	34	6	Multivariate	Categorical, Integer	1998
Iris	150	4	3	Multivariate	Real	1998
Ecoli	336	8	8	Multivariate	Real	1996
Wine	178	13	3	Multivariate	Integer, Real	1991
Glass	214	9	7	Multivariate	Real	1987
Thyroid	215	5	3	Multivariate	Categorical, Real	1987
Vowel	990	10	11	Multivariate	Real	-
Balance	625	4	3	Multivariate	Categorical	1994
Yeast	1484	8	10	Multivariate	Real	1996
Satimage	6435	36	7	Multivariate	Integer	1993
Letter	20000	16	26	Multivariate	Integer	1991
Pendigits	10992	16	10	Multivariate	Integer	1998
Segmentation	2310	19	7	Multivariate	Real	1990
OptDigits	5620	64	10	Multivariate	Integer	1998
Shuttle	14500	9	7	Multivariate	Integer	1993
Vehicle	846	18	4	Multivariate	Integer	-

Appendix G

Publications

G.1 Journals

Sergio Escalera, Oriol Pujol, and Petia Radeva. Boosted Landmarks of Contextual Descriptors and Forest-ECOC: a novel framework to detect and classify objects in cluttered scenes. In Pattern Recognition Letters, vol 28/13, pp. 1759-1768, 2007.

Oriol Pujol, Sergio Escalera, and Petia Radeva. Optimal Node Embedding in Error Correcting Output Codes. In Pattern Recognition, vol. 14, issue 2, febrero 2008, pp. 713-725.

Sergio Escalera, David Tax, Oriol Pujol, Petia Radeva, and Robert P.W. Duin, Subclass Problem-dependent Design of Error-Correcting Output Codes. In Pattern Analysis and Machine Intelligence, vol. 30, issue 6, pp. 1041-1054, 2008. (JCR: 4,306).

Sergio Escalera, Oriol Pujol, and Petia Radeva, Complex Salient Regions for Computer Vision Problems. In Journal on Advances in Signal Processing EURASIP, in press.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Traffic Sign Recognition System with β-Correction. In Machine Vision and Applications, in press.

Sergio Escalera, Oriol Pujol, Josepa Mauri, and Petia Radeva, Intravascular Ultrasound Tissue Characterization with Sub-class Error-Correcting Output Codes Article. In Special Issue on Biomedical Imaging, Journal of Signal Processing Systems, in press.

Xavier Baró, Sergio Escalera, Jordi Vitrià, Oriol Pujol, and Petia Radeva, Traffic Sign Recognition using Evolutionary Adaboost detection and Forest-ECOC classification. In IEEE Transactions on Intelligent Transportation Systems, in press.

Sergio Escalera, Oriol Pujol, and Petia Radeva, On the Decoding Process in Ternary Error-Correcting Output Codes. In Pattern Analysis and Machine Intelligence, second revision.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Separability of Ternary Codes for Sparse Designs of Error-Correcting Output Codes. In Pattern Recognition Letters, second revision.

G.2 Conferences and Workshops

S. Escalera, and Petia Radeva. Fast greyscale model matching and recognition. In Recent Advances in Artificial Intelligence Research and Development. J. Vitrià et.al. (Eds.) IOS Press, pp.69-76, 2004.

Sergio Escalera, Oriol Pujol, Petia Radeva, Optimal extension of Error Correcting Output Codes. In Congrès Català en Intelligència Artificial, 2006, pp. 28-36, Perpignan, Francia, 2006.

Sergio Escalera, Oriol Pujol, Petia Radeva, Decoding of ternary Error Correcting Output Codes. In Iberoamerican Congress on Pattern Recognition CIARP06, Lecture Notes in Computer Science 4225, pp. 753-763, Cancún, Méjico, 2006.

Sergio Escalera, Oriol Pujol, Petia Radeva, Problem-dependent Designs for Error Correcting Output Codes. In Computer Vision Center International Workshop, pp. 13-17, Barcelona, 6 October 2006.

Sergio Escalera, Oriol Pujol and Petia Radeva. ECOC-ONE: A novel coding and decoding strategy. In International Conference on Pattern Recognition ICPR'06, vol. 3, pp. 578-581, Hong Kong, 2006.

Sergio Escalera, Oriol Pujol and Petia Radeva. Forest Extension of Error Correcting Output Codes and Boosted Landmarks. In International Conference on Pattern Recognition ICPR'06, vol.4, pp. 104-107, Hong Kong, 2006.

Sergio Escalera, Oriol Pujol, Petia Radeva, Robust Complex Salient Regions. In Iberian Conference on Pattern Recognition and Image Analysis IbPRIA, Lecture Notes in Computer Science n. 4478, vol. 2, pp. 113-121, 2007.

Alicia Fornés, Sergio Escalera, Josep Lladós, Gemma Sánchez, Petia Radeva, and Oriol Pujol, Handwritten Symbol Recognition by a Boosted Blurred Shape Model with Error Correction. In Iberian Conference on Pattern Recognition and Image Analysis IbPRIA, Lecture Notes in Computer Science n. 4477, vol. 1, pp. 13-21, 2007.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Traffic Sign Recognition using Er-

ror Correcting Techniques. In International Conference on Computer Vision Theory and Applications VISAPP 2007, pp. 281-285, 2007.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Complex Salient Regions for Computer Vision Problems. In IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007.

Alicia Fornés, Sergio Escalera, Josep Lladós i Gemma Sánchez, Symbol Recognition by Multi-class Blurred Shape Models. In 7th IAPR International Workshop on Graphics Recognition, Brazil, Grec 2007.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Loss-Weighted Decoding for Error-Correcting Output Codes, Computer Vision. In Advances in Research and Development, Proceedings of the 2nd CVC Workshop, pp. 77-82, October 2007.

Sergio Escalera, Alicia Fornés, Oriol Pujol, Josep Lladós, and Petia Radeva, Multi-class Binary Object Categorization using Blurred Shape Models. In Iberoamerican Congress on Pattern Recognition CIARP 2007.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Loss-Weighted Error-Correcting Output Codes. In International Conference on Computer Vision Theory and Applications VISAPP, vol. 2, pp. 117-122, Madeira, 2008.

Sergio Escalera, Oriol Pujol, and Petia Radeva, Sub-class Error-Correcting Output Codes. In International Conference on Vision Systems ICVS, LNCS 5008, pp. 494-504, 2008.

Ernest Valveny, Philippe Dosch, Alicia Fornés, and Sergio Escalera, Report on the Third Contest on Symbol Recognition. In Grec 2008.

Alicia Fornés, Sergio Escalera, Josep Lladós, Gemma Sánchez, and Joan Mas, Hand Drawn Symbol Recognition by Blurred Shape Model descriptor and a Multi-class Classifier. In Grec 2008.

Sergio Escalera, Oriol Pujol, Eric Laciar, Jordi Vitrià, Esther Pueyo, and Petia Radeva, Coronary Damage Classification of Patients with the Chagas Disease with Error-Correcting Output Codes. In Intelligent Systems 2008.

Sergio Escalera, Oriol Pujol, Josepa Mauri, and Petia Radeva, IVUS Tissue Characterization with Sub-class Error-Correcting Output Codes. In Mathematical Methods for Biomedical Image Analysis, Computer Vision and Pattern Recognition, 2008.

G.3 Technical Reports

Sergio Escalera. Fast traffic model matching and recognition on gray-scale images. In Computer Vision Center Technical Report #84, CVC(UAB), 2005.

Bibliography

[1] http://www.cis.temple.edu/latecki/research.html.

[2] http://www.vision.caltech.edu/html-files/archive.html.

[3] Prtools toolbox. http://www.prtools.org/, 2007. Faculty of Applied Physics, Delft University of Technology, The Netherlands.

[4] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, transactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[5] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. volume 1, pages 113–141, 2002.

[6] J. Amores, N. Sebe, and P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. *CVPR*, 2005.

[7] J. Amores, N. Sebe, and P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. In *CVPR*, volume 2, pages 769–774, 2005.

[8] A. Asuncion and D.J. Newman. UCI machine learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007. University of California, Irvine, School of Information and Computer Sciences.

[9] S. Belongie, J. Malik, , and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, citeseer.ist.psu.edu/belongie00shape.html, 2000.

[11] A. Bovik, M. Clark, and W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

[12] G. Breithardt, M. E. Cain, N. El-Sherif, N. C. Flowers, V. Hombach, M. Janse, M. B. Simson, and G. Steinbeck. Standards for analysis of ventricular late potentials using high-resolution or signal-averaged electrocardiography. *Circulation*, 83:1481–1488, 1991.

[13] A. P. Burke, A. Farb, G. T. Malcom, J. Smialek, and R. Virmani. Coronary risk factors and plaque morphology inmen with coronary disease who died suddently. *The New England Journal of Medicine*, 336:1276–1281, 1997.

[14] H. Carrasco, D. Jugo, R. Medina, C. Castillo, and P. Miranda. Electrocardiograma de alta resolución y variabilidad de la frecuencia cardiaca en pacientes chagásicos crónicos. *Arch. Inst. Cardiol. Mex.*, 67:277–285, 1997.

[15] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[16] J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A.Serra, and J.Talaya. On the accuracy and performance of the geomobil system. In *International Society for Photogrammetry and Remote Sensing*, 2004.

[17] K. Crammer and Y. Singer. On the learnability and design of output codes for multi-class problems. In *Machine Learning*, volume 47, pages 201–233, 2002.

[18] GREC2005 database. http://symbcontestgrec05.loria.fr/formatgd.php.

[19] GREC2007 database. http://www.epeires.org/.

[20] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(A):1160–1169, 1985.

[21] H. Daume and D. Marcu. A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, 6:1551–1577, 2005.

[22] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[23] L. Devroye, L. Gyorfi, and G. Lugosi. A probabilistic theory of pattern recognition. In *Berlin: Springer-Verlag*, 1996.

[24] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. In *Journal of Artificial Intelligence Research*, volume 2, pages 263–286, 1995.

[25] T. Dietterich and E. Kong. Error-correcting output codes corrects bias and variance. In Proceedings of the 21th International Conference on Machine Learning, editor, *S. Prieditis and S. Russell*, pages 313–321, 1995.

[26] L. Dopico, J. Nadal, and A. Infantosi. Analysis of late potentials in the high-resolution electrocardiogram of patients with chagas disease using weighted coherent average. *Revista Brasileira de Engenharia Biomédica*, 16(1):49–59, 2000.

[27] R.O. Duda and P.E. Hart. Pattern classification and scene analysis. In *New York: John Wiley & Sons*, 1973.

[28] R. Duin and E. Pekalska. The science of pattern recognition; achievements and perspectives. In W. Duch and J. Mandziuk, editors, *Challenges for Computational Intelligence, Studies in Computational Intelligence*, volume 63, pages 221–259, 2007.

[29] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Madison*, 2003.

[30] R. Fergus and A. Zisserman. Object class recognition by unsupervised scaleinvariant learning. *CVPR*, 2003.

[31] A. Fornés, J. Lladós, and G. Sánchez. Primitive segmentation in old handwritten music scores. *Graphics Recognition: Ten Years Review and Future Perspectives*, 3926:279–290, 2006.

[32] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Dept. of Statistics, Stanford University Technical Report., 1998.

[33] R. Ghani. Combining labeled and unlabeled data for text classification with a large number of categories. pages 597–598, 2001.

[34] D. Gil, A. Hernandez, O. Rodrguez, F. Mauri, and P. Radeva. Statistical strategy for anisotropic adventitia modelling in ivus. *IEEE Trans. Medical Imaging*, 27:1022–1030, 2006.

[35] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1999.

[36] P. Hong and T. Huang. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relations graphs. *International workshop on Combinational Image Analysis*, 139:113–135, 2004.

[37] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification. *Department of CSIE, technical report*, 2002.

[38] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, 2000.

[39] J.N. Kapur and H.K. Kesavan. Entropy optimization principles with applications. *Academic press, London*, 1992.

[40] C. Karla, B. Joel, P. Oriol, N. Salvatella, and P. Radeva. In-vivo ivus tissue classification: a comparison between rf signal analysis and reconstructed images. *Progress in Pattern Recognition*, pages 137–146, 2006.

[41] M. Kawasaki. In vivo quantitative tissue characterization of human coronary arterial plaques by use of integrated backscatter intravascular ultrasound and comparison with angioscopic findings. *Circulation*, 105:2487–2492, 2002.

[42] T. Kikuchi and S. Abe. Error correcting output codes vs. fuzzy support vector machines. In *ANNPR*, 2003.

[43] W. Kim. A new region-based shape descriptor. *Technical report, Hanyang University and Konan Technology*, 1999.

[44] J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face verification using error correcting output codes. *CVPR*, 1:755–760, 2001.

[45] E. Laciar, R. Jané, and D. H. Brooks. Evaluation of myocardial damage in chagasic patients from the signal-averaged and beat-to-beat analysis of the high resolution electrocardiogram. *Computers in Cardiology*, 33:25–28, 2006.

[46] E. Laciar, R. Jané, D. H. Brooks, and A. Torres. Análisis de senal promediada y latido a latido del ecg de alta resolución en pacientes con mal de chagas. *XXIV Congreso Anual de la Sociedad Espanola de Ingeniería Biomédica*, pages 169–172, 2006.

[47] J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol recognition: Current advances and perspectives. *Graphics Recognition: Algorithms and Applications*, 2390:104–127, 2002.

[48] Logitech. IO digital pen. www.logitech.com, 2004.

[49] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110, 2004.

[50] G. Loy and A. Zelinsky. A fast radial symmetry transform for detecting points of interest. volume 25, pages 959–973, 2003.

[51] J. H. Maguire, R. Hoff, I. Sherlock, A. C. Guimaraes, A. C. Sleigh, N. B. Ramos, K. E. Mott, and T. H Séller. Cardiac morbidity and mortality due to chagas disease: prospective electrocardiographic study of a brazilian community circulation. 75:1140–1145, 1987.

[52] B. Manjunath, P. Salembier, and T. Sikora. Introduction to mpeg-7. *Multimedia content description interface, John Wiley and Sons*, 2002.

[53] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna. A wavelet-based ecg delineator: evaluation on standard databases. *IEEE Trans. Biomed. Eng.*, 51:570–581, 2004.

[54] K. Mikolajczyk and C. Schmid. Scale nd affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.

[55] F. Mora, P. Gomis, and G. Passariello. Senales electrocardiográficas de alta resolución en chagas. *El proyecto SEARCH Acta Científica Venezolana*, 50:187–194, 1999.

[56] B. S. Morse. Segmentation (edge based, hough transform). Brigham Young University, 2000.

[57] Standard MPEG. ISO/IEC 15938-5:2003(E).

[58] A. Murashige, T. Hiro, T. Fujii, K. Imoto, T. Murata, Y. Fukumoto, and M. Matsuzaki. Detection of lipid-laden atherosclerotic plaque by wavelet analysis of radiofrequency intravascular ultrasound signals. *Journal of the American College of Cardiology*, 45:1954–1960, 2005.

[59] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: A graphical model relating features, objects, and scenes. *Advances in NIPS*, 2003.

[60] A. Nair, B. Kuban nd N. Obuchowski, and G. Vince. Assesing spectral algorithms to predict atherosclerotic plaque composition with normalized and raw intravascular ultrasound data. *Ultrasound in Medicine & Biology*, 27:1319–1331, 2001.

[61] J. Matas nd O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Proceedings of the British Machine Vision Conference*, 1:384–393, 2002.

[62] WHO Division of Control of Tropical Diseases. Chagas disease elimination. burden and trends. *WHO web site www.who.int/ctd/html/chagburtre.html*.

[63] P. Ohanian and R. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25:819–833, 1992.

[64] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[65] World Health Organization. World health organization statistics. *http://www.who.int/entity/healthinfo/statistics/*, 2006.

[66] OSU-SVM-TOOLBOX. http://svm.sourceforge.net/.

[67] A. Passerini, M. Pontil, and P. Frasconi. New results on error correcting output codes of kernel machines. In *IEEE Transactions on Neural Networks*, volume 15, pages 45–54, 2004.

[68] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. Learning machines. In *McGraw-Hill*, 1965.

[69] J. Proakis, C. Rader, F. Ling, and C. Nikias. Advanced digital signal processing. *Mc Millan*, 1992.

[70] P. Pudil, F. Ferri, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. *Proc. Int. Conf. Pattern Recognition*, pages 279–283, 1994.

[71] E. Pueyo, E. Anzuola, E. Laciar, P. Laguna, and R. Jané. Evaluation of QRS slopes for determination of myocardial damage in chronic chagasic patients. *Computers in Cardiology*, 2007.

[72] E. Pueyo, L. Sornmo, and P. Laguna. QRS slopes for detection and characterization of myocardial ischemia. *IEEE Trans. Biomed. Eng.*, 2007.

[73] O. Pujol, P. Radeva, and J. Vitrià. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. In *Trans. on PAMI*, volume 28, pages 1001–1007, 2006.

[74] O. Pujol, M. Rosales, and P. Radeva. Intravascular ultrasound images vessel characterization using adaboost. *Functional Imaging and Modelling of the Heart*, pages 242–251, 2003.

[75] E. Punset. El alma está en el cerebro. In *Aguilar ed.*, 2006.

[76] P. Perona R. Fergus and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[77] T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:291–310, 1999.

[78] F. Ricci and D. Aha. Error-correcting output codes for local learners. *European conference on machine learning*, 1398:280–291, 1998.

[79] M. Riesenhuber and T. Poggio. Computational models of object recognition in cortex: A review. In *Technical report, MIT AI laboratory, Center for Biological and Computational Learning*, 2000.

[80] G. Sànchez, E. Valveny, J. Lladós, J. Mas, and N. Lozano. A platform to extract knowledge from graphic documents. application to an architectural sketch understanding scenario. *IAPR Workshop on Document Analysis Systems (DAS2004)*, 3163:389–400, 2004.

[81] F. Schafialitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92(2):236–264, 2003.

[82] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated prediction. In *Machine Learning*, volume 37, pages 297–336, 1999.

[83] C. Schmid and R. Mohr. Local gray value invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[84] H. Schneiderman. Learning a restricted bayessian network for object detection. *CVPR*, 2004.

[85] Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[86] P. Simard, Y. LeCum, J. Denker, and B. Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. *Neural Networks: Tricks of the Trade*, 1524:239–274, 1998.

[87] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *Proceedings of the International Conference on Computer Vision*, 2003.

[88] T.Hastie and R.Tibshirani. Classification by pairwise grouping. *NIPS*, 26:451–471, 1998.

[89] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *JMLR*, 3:1415–1438, 2003.

[90] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. In *CVPR*, volume 2, pages 762–769, 2004.

[91] W. Utschick and W. Weichselberger. Stochastic organization of output codes in multiclass learning problems. In *Neural Computation*, volume 13, pages 1065–1102, 2004.

[92] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–511–I–518, 2001.

[93] Joachim Weickert. *Anisotropic diffusion in image processing.* ECMI Series. Teubner-Verlag, Stuttgart, 1998.

[94] T. Windeatt and G. Ardeshir. Boosted ecoc ensembles for face recognition. *International Conference on Visual Information Engineering*, pages 165–168, 2003.

[95] T. Windeatt and R. Ghaderi. Coding and decoding for multi-class learning problems. In *Information Fusion*, volume 4, pages 11–21, 2003.

[96] Q. Zgu. Minimum cross-entropy approximation for modeling of highly intertwining data sets at subclass levels. *Journal of Intelligent Information Systems*, 11:139–152, 1998.

[97] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.

[98] W. Zhang, L. Wenyin, and K. Zhang. Symbol recognition with kernel density matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2020–2024, 2006.

[99] X. Zhang, C. R. McKay, and M. Sonka. Tissue characterization in intravascular ultrasound images. *IEEE Transactions on Medicine*, 17:889–898, 1998.

[100] J. Zhou and C. Suen. Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach. volume 1, pages 484–488, 2005.

[101] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class adaboost. a multiclass generalization of the adaboost algorithm, based on a generalization of the exponential loss. 2005.

[102] M. Zhu and A. M. Martinez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1274–1286, 2006.