Learning error-correcting representations for multi-class problems

Miguel Ángel Bautista Martín

Universitat de Barcelona

Learning Error-Correcting Representations for Multi-Class Problems

A dissertation submitted by **Miguel Ángel Bautista Martín** at Universitat de Barcelona to fulfil the degree of **Doctor en Matemàtiques**.

Barcelona, November 25, 2015

Director
Dr. Sergio Escalera
Dept. Matemàtica Aplicada i Análisi
Universitat de Barcelona

Co-director
Dr. Oriol Pujol
Dept. Matemàtica Aplicada i Análisi
Universitat de Barcelona

This document was typeset by the author using LaTeX 2_{ε}.

The research described in this book was carried out at Dept. Matemàtica Aplicada i Análisi, Universitat de Barcelona and the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN: XXXXX

Printed by Ediciones Gráficas Rey, S.L.

A mis padres y hermana de los que aprendo cada día.

Acknowledgements

As most things in life, this dissertation is the result of certain people allocated in certain time and place, I would like to thank all of them here. First and foremost I am deeply thankful to my supervisors Dr. Sergio Escalera and Dr. Oriol Pujol for their motivation, hard work and efforts during all these years, this would have not been possible without them. I would also like to thank Dr. Xavier Baró for his guidance and comments.

Staying at the Human Sensing group Carnegie Mellon University I had the chance to meet Dr. Fernando de la Torre to whom I am very thankful for sharing his experience, knowledge and overall working process with me. I am also very thankful to the bright students that I met there and with whom I enjoyed a lot of very nice discussions. Amongst them all, I would like to thank Ricardo, Paco and Ishan for sharing their comments and experiences with me. I also would like to thank to all the people at the Applied Mathematics and Analysis department at University of Barcelona: Francesco, Piero, Adriana, Víctor, Jordi, Petia, Eloi y Laura. All of you shared your moments, experiences and discussions with me, making me a better scientist.

Engaging in the birth of the Human Pose and Behavior Analysis (HuPBA) group at the University of Barcelona, was one of the best scientific experiences I have ever had. I am deeply thankful to all my colleagues at the HuPBA group. In particular, to Toni, Xavi, Victor and Albert. Not only did I learn from their hard-work, dedication and knowledge but I also shared with them great memories that I will hold dear forever.

Levemente, quisiera tomarme unas líneas en español para agradecer a todos mis amigos en Ibiza y Barcelona que durante estos años han estado a mi lado haciendo que el camino fuera más fácil. En primer lugar querría agradecer a Pau los momentos que hemos compartido juntos durante todos estos años. También me gustaría agradecer a Adri, Adrián y Javi por los años compartidos en Barcelona. A todas las nuevas amistades hechas en Barcelona: Jordi, Óscar, Christian, Daniel, David, Vicente, Borja, Víctor, Oriol, y un largo etcétera que ocuparía varias páginas. Finalmente, quiero agradecer a mis padres Ana y Miguel por la educación que me han dado así como por los valores de humildad, esfuerzo y honestidad que siempre me han inculcado.

A mi hermana Ana quiero dedicarle no solo esta línea sino la tesis en su totalidad, es un ejemplo para mí.

Two monks were watching a flag flapping in the wind. One said to the other, "The flag is moving." The other replied, "The wind is moving." the sixth and last patriarch of Chán Buddhism who was happening to pass by, overheard this. He said, "Not the flag, not the wind; mind is moving."

Case 29 - *The Gateless Gate*

Abstract

Real life is full of multi-class decision tasks. In the Pattern Recognition field, several methodologies have been proposed to deal with binary problems obtaining satisfying results in terms of performance. However, the extension of very powerful binary classifiers to the multi-class case is a complex task. The Error-Correcting Output Codes framework has demonstrated to be a very powerful tool to combine binary classifiers to tackle multi-class problems. However, most of the combinations of binary classifiers in the ECOC framework overlook the underlaying structure of the multi-class problem. In addition, is still unclear how the Error-Correction of an ECOC design is distributed among the different classes.

In this dissertation, we are interested in tackling critic problems of the ECOC framework, such as the definition of the number of classifiers to tackle a multi-class problem, how to adapt the ECOC coding to multi-class data and how to distribute error-correction among different pairs of categories.

In order to deal with this issues, this dissertation describes several proposals. 1) We define a new representation for ECOC coding matrices that expresses the pair-wise codeword separability and allows for a deeper understanding of how error-correction is distributed among classes. 2) We study the effect of using a logarithmic number of binary classifiers to treat the multi-class problem in order to obtain very efficient models. 3) In order to search for very compact ECOC coding matrices that take into account the distribution of multi-class data we use Genetic Algorithms that take into account the constraints of the ECOC framework. 4) We propose a discrete factorization algorithm that finds an ECOC configuration that allocates the error-correcting capabilities to those classes that are more prone to errors.

The proposed methodologies are evaluated on different real and synthetic data sets: UCI Machine Learning Repository, handwriting symbols, traffic signs from a Mobile Mapping System, and Human Pose Recovery. The results of this thesis show that significant performance improvements are obtained on traditional coding ECOC designs when the proposed ECOC coding designs are taken into account.

Resumen

En la vida cotidiana las tareas de decisión multi-clase surgen constantemente. En el campo de Reconocimiento de Patrones muchos métodos de clasificación binaria han sido propuestos obteniendo resultados altamente satisfactorios en términos de rendimiento. Sin embargo, la extensión de estos sofisticados clasificadores binarios al contexto multi-clase es una tarea compleja. En este ámbito, las estrategias de Códigos Correctores de Errores (CCEs) han demostrado ser una herramienta muy potente para tratar la combinación de clasificadores binarios. No obstante, la mayoría de arquitecturas de combinación de clasificadores binarios negligen la estructura del problema multi-clase. Sin embargo, el análisis de la distribución de corrección de errores entre clases es aún un problema abierto.

En esta tesis doctoral, nos centramos en tratar problemas críticos de los códigos correctores de errores; la definición del numero de clasificadores necesarios para tratar un problema multi-clase arbitrario; la adaptación de los problemas binarios al problema multi-clase y cómo distribuir la corrección de errores entre clases. Para dar respuesta a estas cuestiones, en esta tesis doctoral describimos varias propuestas. 1) Definimos una nueva representación para CCEs que expresa la separabilidad entre pares de códigos y nos permite una mejor comprensión de cómo se distribuye la corrección de errores entre distintas clases. 2) Estudiamos el efecto de usar un numero logarítmico de clasificadores binarios para tratar el problema multi-clase con el objetivo de obtener modelos muy eficientes. 3) Con el objetivo de encontrar modelos muy eficientes que tienen en cuenta la estructura del problema multi-clase utilizamos algoritmos genéticos que tienen en cuenta las restricciones de los ECCs. 4) Proponemos un algoritmo de factorización de matrices discreta que encuentra ECCs con una configuración que distribuye corrección de error a aquellas categorías que son más propensas a tener errores.

Las metodologías propuestas son evaluadas en distintos problemas reales y sintéticos como por ejemplo: Repositorio UCI de Aprendizaje Automático, reconocimiento de símbolos escritos, clasificacion de señales de trafico y reconocimiento de la pose humana. Los resultados obtenidos en esta tesis muestran mejoras significativas en rendimiento comparados con los diseños tradiciones de ECCs cuando las distintas propuestas se tienen en cuenta.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The quintessential goal of Artificial Intelligence is to build systems that replicate (or even improve) the processes performed by humans in order to acquire knowledge and interpret the environment around them. Although the Artificial Intelligence field has recently seen enormous advances, specially in those areas regarding with acquisition of knowledge from perception (e.g Computer Vision, Speech Recognition, Remote Sensing, etc.), we are still far from reaching human performance in different tasks.

This is due to the fact that humans spend an incredible number of hours gathering information from which make hypotheses, which become more credible as the quantity of information acquired increases. This phenomenon is clearly observed in babies (see Figure 1.1), which spend few years perceiving their environment by looking, touching, hearing, smelling and asking an innumerable number of questions. This acquisition of information by means of the sensory stimuli and supervision, leads to an improvement on the confidence of their hypotheses. This process is referred to as *learning*. In fact, the problem of taking decisions among different hypotheses given sensory stimuli is often described as supervised categorization.

This dissertation tackles the problem of modeling different hypotheses to solve multi-class pattern and object categorization problems.

Figure 1.1: Baby analyzing shapes and colors.

1.1 Motivation

Humans perceive and reason about the world that surrounds them from multi-modal stimuli that sensory organs capture. Even though those stimuli are of different nature and therefore come from different sensory organs (e.g. light stimuli is captured by the retina in our eyes while sound waves are captured by the cochlea in our ears), the brain is able to process and integrate all this information. The mammalian brain has been studied over years, yielding interesting findings on its topological organization. As a result, it has been observed that the cerebral cortex can be categorized in three kinds of regions:

- Motor areas, which are dedicated to move parts of the body.
- Sensory areas, dedicated to process the input signals captured by sensory organs. For example, the V1 area (primary visual cortex) is specialized in recognizing patterns coming from visual stimuli, while the A1 area (primary auditory cortex) is specialized in processing sounds captured by the ears. In addition, some smaller regions have been discovered to be dedicated to much specific tasks, e.g. the Fusiform Face Area [78] for facial recognition, the Parahippocampal Place Area [44] for scene context recognition, or the Extrastriate Body Area [40] for recognizing parts of the human body.
- Association areas, produce meaningful perceptual experience of the world by integrating sensory information and information stored in memory. As pointed out by [140], association areas are globally organized as distributed networks, connecting different widely-spaced parts of the human brain. More specifically, the authors found a coarse parcellation of the brain in seven different regions (and a finer categorization in 17 regions), which can be connected following different pathways. While it is clear that our brains integrate signals and hypotheses coming from different sensory modalities, for example, when trying to perform **recognition** a word from its sound and the movement of the lips of the speaker, it still remains unclear how this integration process is exactly performed. Among the different theories, statistical integration has been backed by a number or works providing results that correlate with human behaviour [13, 92, 134].

Given that different sensory modalities are not equally reliable (e.g. vision is usually more reliable than hearing in daylight, but not at night), it is natural to think of a statistical approach or a combination of statistical approaches to combine multi-modal cues coming from different senses. Hence, it has been shown that in order to improve the confidence of the decision making process the human brain uses information coming from different sensory areas in the brain [140]. This has three obvious reasons:

- The decision taking capabilities of the brain, although great are limited and can be defective in certain situations.
- The information acquired by the sensory areas in the brain can be imperfect, due to environmental conditions or faulty processing abilities.
- The experience stored in memory only represents a part of the space of the information and possible decisions available for the task.

Therefore, by using different sources of information the brain is able to make more confident decisions with a capability to generalize to different tasks.

However, studies have shown that the reasoning processes related to our visual system play a very important role in our global intelligent behavior. This can be often proved by trying to interact with our daily environment without perceiving visual information. Being deprived of this information makes simple tasks (i.e walking, avoiding obstacles, etc.) much harder. The area of Artificial Intelligence that deals with the processing of visual information is known as Computer Vision, and it covers from the extraction valuable information from digital two-dimensional projections of the real world, up to how process that information to obtain high level representations to understand and make sense of what is happening in those images.

For this reason, in this dissertation we are particularly interested in categorization problems which deal with visual data as their main source of information.

1.2 Statistical Pattern Recognition

Automatic (machine) recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, artificial intelligence, and remote sensing. But what is a pattern? Watanabe [132] defines a pattern as the opposite of chaos; it is an entity, that could be given a name and which exhibits a certain type of structure. For example, a pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. Given a pattern, its recognition/classification may consist of one of the following tasks: 1) Supervised Classification (categorization) in which the input pattern is identified as a member of a predefined class, 2) Unsupervised Classification (clustering) in which the pattern is assigned to a hitherto unknown class. Note that the recognition problem here is being posed as a classification or categorization task, where the classes are either defined by the system designer (categorization) or are learned based on the similarity of patterns (clustering). 3) Semi-supervised learning is halfway between supervised and unsupervised learning. In addition to patters without any predefined classes, the algorithm is also provided with some patterns with their predefined classes – but not necessarily for all examples.

Statistical Pattern Recognition denotes as **Supervised Classification** the problem of making a decision based on certain learning information available. [76]. In this sense, a classification system looks for a method that makes those judgments or decisions in previously unseen situations. In particular, a binary classification task refers to the problem of making a decision for a new object $\mathbf{x}_i$ (data sample or pattern), so that ξ_i is classified among two predefined categories (or classes), c^1 and c^2. Thus, in binary classification we only have two alternatives to decide amongst c^1 and c^2. Formally, given training data $\{\mathbf{X}, \mathbf{y}\}$ $\mathbf{X} \in \mathbb{R}^{n \times m}, \mathbf{y} \in \{c^1, c^2\}^{1 \times n}$, where $\mathbf{x}_i$ is the $i-$th object and y_i is the true class of object $\mathbf{x}_i$, a classifier function f is trained to distinguish the objects which label is c^1 from the objects which label is c^2. The classifier function is defined as $f : \mathbf{X} \to \mathbf{y}$, where $y_i \in \{c^1, c^2\} \forall i \in \{1, \ldots, n\}$. However, in order to approximate or learn the classifier f each element $\mathbf{x}_i$ should be described with a set of characteristic or features, inherent to the object itself. For instance, we could describe a baby based on his weight, height, eyes color, etc. Then, the process of approximating the classifier function also called learning or training uses the set of object features from all the samples in order to define a boundary between two classes.

There are different techniques that deal with the task of object/pattern description [14][91]. Informative features depends on the object itself and its relationships with other classes, as well as the problem one wants to solve. In addition, some features can change their description when environmental factors introduce noise in the description process. For example, using the weight as feature description of a baby can be sensible to the planet in which this measure is being performed. In particular, when describing or extracting features from images, we can find that such features are sensible to changes in illumination, occlusions between parts or deformations on non-rigid entities.

The problem of object description is a very difficult task. Observe the objects of Figure 1.2. Which are the representative features to describe a chair? shape? color? Obviously, it depends on the categorization problem we consider. Binary classification not only does consist on distinguishing apples from tables. In the chair class, we can also apply categorization. Which chairs are broken? Which chairs are black or brown? 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.

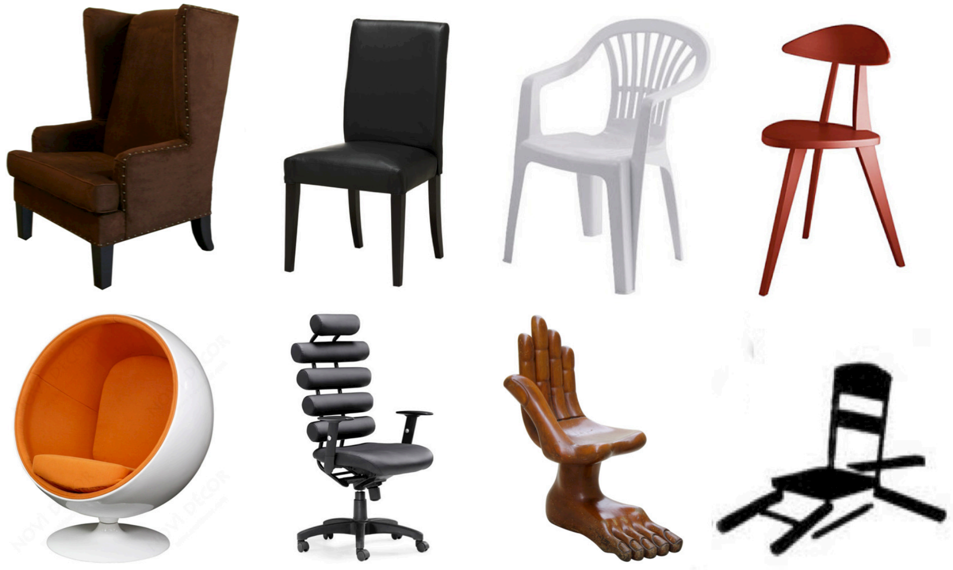


Figure 1.2: Chair samples.

Given the features or descriptions obtained for each data sample, in the statistical approach, each of these patterns is viewed as a point in a high-dimensional space of features. 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 [17]. 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 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 [17][129].

The classification or categorization system is operated in two modes: learning (training) and classification (testing) (see Figure 1.3). The role of the preprocessing module is to remove noise, normalize the pattern, and any other operation which will contribute in defining a compact representation of the data samples. 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 [42]:

- How the learning system should adapt to the representation?: 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 description, or it may enrich it by emphasizing 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 data samples. 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 Figure 1.3.

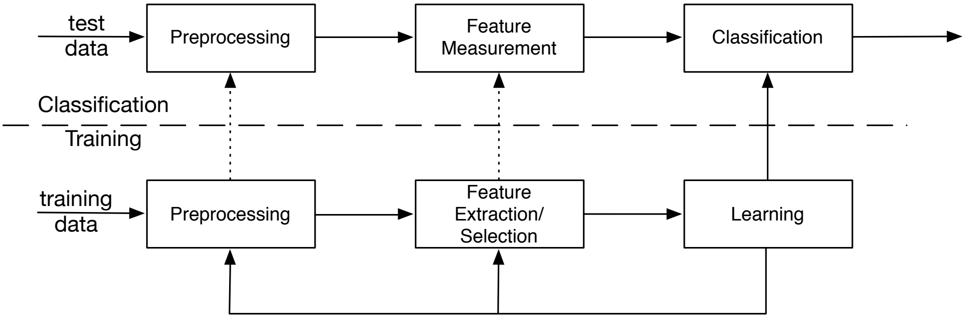


Figure 1.3: Standard pipeline for statistical pattern recognition.

This dissertation aims to focus on a subset of the previous question to present powerful multi-class pattern and object recognition systems. The architecture of multi-class strategies that we present adapt the previous representation of the data samples, in a problem-dependent way, so that the learning process obtains high generalization performance.

1.2.1 Visual Pattern Recognition

One of the most challenging areas in which statistical pattern recognition and learning theory is applied is the field of Computer Vision. Many of the visual Pattern Recognition techniques that achieve current state-of-the-art results are biological inspired [121]. The majority of studies in this area assume that not all parts of an image give us valuable information, and only analyzing the most important parts of the image in detail is sufficient to perform recognition or categorization tasks. 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 facades. 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 focus 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 1.4.

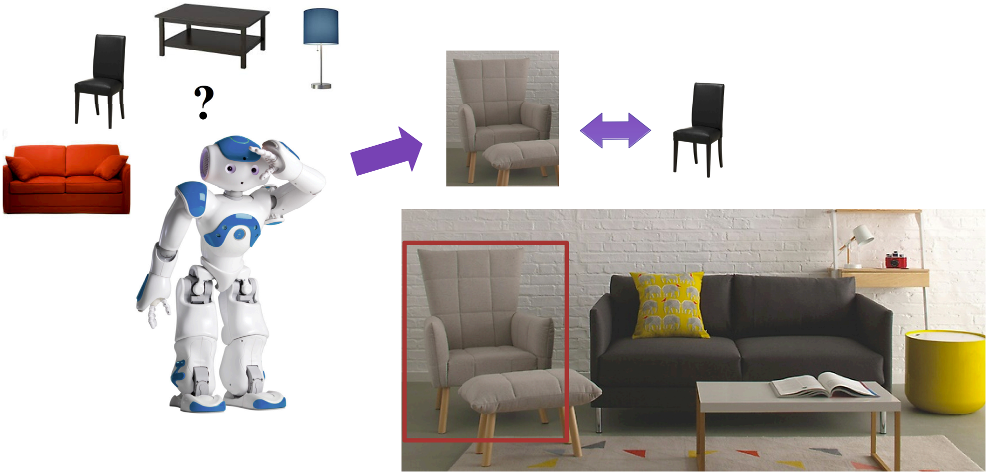


Figure 1.4: Nao detects and classifies the furniture in the scene.

The humanoid robot Nao from Aldebaran Robotics captures images from a scene, discarding background regions. At the first step, it treats to find the regions of the image that contain a piece of furniture. Once the region containing a piece of furniture is found, given four previous furniture categories, Nao classifies the inner object as a chair. Hence, the problem of object recognition can be seen as an object detection problem followed by a classification procedure.

1.3 The Multi-class Classification Problem

When using the term binary classification, the labels from classes c^1 and c^2 use to take the values $+1$ or -1, respectively as a convention. At the learning process explained in previous sections, the labels for the training objects are known.

In previous examples, the underlaying problem corresponded to Supervised Binary Classification. However, Binary Classification is far from representing real world problems. We can classify between black and brown chairs or we can also distinguish among chairs, tables, couches, lamps, 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 k possible categories (obviously, the number of objects that we have learnt during our life tends to be uncountable). If we can design a multi-classifier F, then the prediction can be understood as in the binary classification problem, being $F : \mathbf{X} \rightarrow \mathbf{y}$, where now $y_i \in \{1, \ldots, k\}$, for a $k-$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 in successive sections, a possible multi-class solution could potentially consist of designing and combining of a set of binary classification problems.

1.4 State-of-the-art Classification Techniques in Visual Pattern Recognition

In this chapter we describe the state-of-the-art techniques for classification with a particular interest in those applied in Computer Vision.

1.4.1 Classifiers

In the Statistical Pattern Recognition field, classifiers are frequently grouped into those based on similarities, probabilities, or geometric information about class distribution [41, 76].

1. 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.

2. 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 $P(y_j)$ is the prior probability that a given instance $\mathbf{x}_j$ belongs to class y_j, $p(\mathbf{x}_j|y_j)$ is the class-conditional probability density function: the density for $\mathbf{x}_j$ given that the instance is of class y_j , and $p(\mathbf{x}_j)$ is defined as $p(\mathbf{x}_j|y_j) \times P(y_j)$ over all classes. Then, Equation 1.1 is equivalent to:

$$P(y_j|\mathbf{x}_j) = \frac{p(\mathbf{x}_j|y_j) \times P(y_j)}{p(\mathbf{x}_j)} \tag{1.2}$$

The classification is done in favor of the j-th class if $P(y_j = i|\mathbf{x}_j) > P(y_j = k|\mathbf{x}_t), \forall i \in \{1, \ldots, k\}$ and $i \neq j$, where $\{1, \ldots, k\}$ is the set of classes.

3. Geometric Classifiers: Geometric classifiers build decision boundaries by directly minimizing an error criterion based on geometric considerations such as the margin, the convex envelope o topological properties of the set.

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.

1.4.2 Multi-class classifiers

Intrinsic multi-class methods

In classification problems the goal is to find a function $F : \mathbf{X} \to \mathbf{y}$, where $\mathbf{X}$ is the set of observations and $\mathbf{y} \in \{1, \ldots, k\}^{1\times n}$ the set composed by the label of each observation ($\max(\mathbf{y}) > 2$ for the multi-class context). The goal of F is to map each observation $\mathbf{x}_i \in \mathbb{R}^{1\times m}$ to its label $y_i \in \{1, \ldots, k\}$. There are many possible strategies for estimating F, nevertheless, literature has shown that the complexity for estimating a unique F for the whole multi-class problem grows with the cardinality of the label set. In this sense, most of the strategies aim to model the probability density function of each category. Moreover, lazy learning methods like Nearest Neighbours to estimate k by a local search of the most proximate observations.

The multi-class problem can be directly treated by some methods that exhibit a multi-class behaviour off the shelf (i.e Nearest Neighbours [133], Decision Trees [107], Random Forests [26]). However, some of the most powerful methods for binary classification like Support Vector Machines (SVM) or Adaptive Boosting (AdaBoost) can not be directly extended to the multi-class case and further development is required. In this sense, literature is prolific on single-loss strategies to estimate F. One of the most well know approaches are the extensions of SVMs [18] to the multi-class case. For instance, the work of Weston and Watkins [135] presents a single-machine extension of the SVM method to cope with the multi-class case, in which k predictor functions are trained, constrained with $k - 1$ slack variables per sample. However, a more recent adaptation of [33] reduces the number of constraints per samples to one, paying only for the second largest classification score among the k predictors. To solve the optimization problem a dual decomposition algorithm is derived, which iteratively solves the quadratic programming problem associated with each training sample. Despite these efforts, single-machine approaches to estimate f scale poorly with the number of classes and are often outperformed by simple decompositions [111, 125]. In recent years various works that extended the classical Adaptive Boosting method [57] to the multi-class setting have been presented [98, 119]. In [147] the authors directly extend the AdaBoost algorithm to the multi-class case without reducing it to multiple binary problems, that is estimating a single F for the whole multi-class problem. This algorithm is based on an exponential loss function for multi-class classification which is optimized on a forward stage-wise additive model. Furthermore, the work of Saberian and Vasconcenlos [116] presents a derivation of a new margin loss function for multi-class classification altogether with the set of real class codewords that maximize the presented multi-class margin, yielding boundaries with max margin. However, though these methods are consistently derived and supported with strong theoretical results, methodologies that jointly optimize a multi-class loss function present some limitations:

- They scale linearly with k, rendering them unsuitable for problems with a large k.
- Due to their single-loss architecture the exploitation of parallelization on modern multi-core processors is difficult.
- They can not recover from classification errors on the class predictors.

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 mayinclude 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
Logisctic 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; non-linear; overtraining insensitive; good generalization performance	Geometric

Table 1.1: Summary of classification methods.

Ensemble Learning for multi-class problems

Multi-class classification (i.e. automatically attributing a label to each sample of the dataset) is one of the classic problems in Pattern Recognition and Machine Intelligence. In this section we review the state-of-the-art for Ensemble Learning techniques that tackle multi-class classification problems.

Divide and Conquer Approaches

The divide and conquer approach has drawn a lot of attention due to its excellent results and easily parallelizable architecture [3, 7, 51, 64, 94, 106, 111, 125]. In this sense, instead of developing a method to cope with the multi-class case, divide and conquer approaches decouple F into a set of l binary problems which are treated separately $F = \{f^1, \ldots, f^l\}$. Once the responses of binary classifiers are obtained a committee strategy is used to find the final output. In this trend one can find three main lines of research: flat strategies, hierarchical classification, and ECOC. Flat strategies like One vs. One [125] and One vs. All [111] are those that use a predefined problem partition scheme followed by a committee strategy to aggregate the binary classifier outputs. Hierarchical classification relies on a similarity metric distance among classes to build a binary tree in which nodes correspond to different problem partitions [60, 64, 94]. Finally, the ECOC framework consists of two steps: In the *coding* step, a set of binary partitions of the original problem are encoded in a matrix of discrete codewords [39] (univocally defined, one codeword per class). At the *decoding* step a final decision is obtained by comparing the test codeword resulting of the union of the binary classifier responses with every class codeword and choosing the class codeword at minimum distance [47, 146]. The coding step has been widely studied in literature, yielding three different types of codings: predefined codings [111, 125], random codings [3] and problem-dependent codings for ECOC [7, 51, 61, 106, 141, 142]. Predefined codings like One vs. All or One vs. One are directly embeddable in the ECOC framework. In [3], the authors propose the Dense and Sparse Random coding designs with a fixed code length of $\{10, 15\}\log_2(k)$, respectively. In [3] the authors encourage to generate a set of 10^4 random matrices and select the one that maximizes the minimum distance between rows, thus showing the highest correction capability. However, the selection of a suitable code length l still remains an open problem.

Problem-dependent designs

Alternatively, problem-dependent strategies for ECOC have proven to be successful in multi-class classification tasks [51, 60, 61, 106, 141, 142, 143, 144]. A common trend of these works is to exploit information of the multi-class data distribution obtained a priori in order to design a decomposition into binary problems that are easily separable. In that sense, [141] computes a spectral decomposition of the graph laplacian associated to the multi-class problem. The expected most separable partitions correspond to the thresholded eigenvectors of the laplacian. However, this approach does not provide any warranties on defining unequivocal codewords (which is a core property of the ECOC coding framework) or obtaining a suitable code length l. In [61], Gao and Koller propose a method which adaptively learns an ECOC coding by optimizing a novel multi-class hinge loss function sequentially. On an update of their earlier work, Gao and Koller propose in [60] a joint optimization process to learn a hierarchy of classifiers in which each node corresponds to a binary subproblem that is optimized to find easily separable subproblems. Nonetheless, although the hierarchical configuration speeds up the testing step, it is highly prone to error propagation since node mis-classifications can not be recovered. In addition, the work of Zhao et. al [142] proposes a dual projected gradient method embedded on a constrained concave-convex procedure to optimize an objective composed of a measure of expected problem separability, codeword correlation and regularization terms. In the light of these results, a general trend of recent works is to optimize a measure of binary problem separability in order to induce easily separable sub-problems. This assumption leads to ECOC coding matrices that boost the boundaries of easily separable classes while modeling with low redundancy the ones with most confusion.

Furthermore, [10] proposed a standard Genetic Algorithm to optimize an ECOC matrix, known as Minimal ECOC matrix, which is the theoretical lower-bound in terms of the number of classifiers $\lceil \log_2 k \ \rceil$. In this work the evaluation of each individual (ECOC matrix) is obtained by means of its classification error over the validation set. In addition, [62] proposed the use of the CHC Genetic Algorithm [52] to optimize a Sparse Random ECOC matrix. In this work, the code length is fixed in the interval $[30, 50]$ independently of the number of classes. Finally, [89] used a Genetic Algorithm to optimize a Sparse Random coding matrix of length in the interval $[\log_2(k), k]$. The evaluation of each individual (ECOC coding matrix) is performed as the classification error over a validation set.

Convolutional Neural Networks

Neural Networks are a family of models inspired by biological neural networks. These models are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on training data, making neural nets adaptive to inputs and capable of learning. Neural Networks (NNs) were successfully applied to multi-class classification problem in 90s [123]. These methods can be seen as a stacking of different neuron layers in which outputs of layer i are the input of layer $i + 1$.

Particularly, the introduction of the multi-layer perceptron by Rosenblatt [113], which was demonstrated to be a universal function approximator [113] made Neural Networks a hot topic at the time. Furthermore, the introduction of back-propagation to train Neural Networks meant that these models could be trained efficiently [59, 113, 115]. However, with the appearance of SVMs in the mid 90s Neural Networks lost all there interest due to the stronger theoretical properties and easier interpretation of the Support Vector Machine model.

In despite of the great performance exihibited by SVMs, several researchers kept on working with Neural Networks after the introduction of SVMs [15, 16, 87, 121]. In this sense, based on previous work from Fukushima [59] Convolutional Neural Networks (CNNs) were first introduced by LeCun and Bengio [86]. The idea of this method was to use a set of convolutional layers to combine three architectural ideas to ensure some degree of shift and distortion invariance: local receptive fields, shared weights and sometimes spatial or temporal subsampling. Each neuron of a layer receives inputs from a set of units located in a small neighborhood in the previous layer.

However, it was not until the paper of Krizhevsky et. al [82] that Convolutional Neural Networks gained an overwhelming attention given the fact that the method proposed by Krizhevsky et. al [82] outperformed other methods by 10% in the ILSRVC challenge [82]. This method followed the line of previous Convolutional Neural Networks [86], defining a deep architecture with 5 convolutional layers and 3 fully-connected layers, generating a model consisting of 60 million parameters. Convolutional Neural Networks have consistently outperformed state-of-the-art methods for most classification tasks in Computer Vision. However, further analyses of this methods on this dissertation falls out of scope given the overly large number of parameters and data needed to train these models.

1.5 Objective of the Thesis

In this dissertation we are interested in proposing novel techniques to learn problem-dependent ECOC coding designs for multi-class problems in Computer Vision, in particular, we are very interest in strategies that scale sub-linearly with the number of classes in the multi-class problem. Hence, we most of our experimental results analyze performance as a function of the complexity of the model.

1. **Reduce the code length of problem-dependent ECOC codings**. Code length has a direct implication in the model complexity of the overall ECOC model. One objective of this dissertation is analyze and explore ECOC coding designs which code length scales sub-linearly with the number of classes in the multi-class problem. Therefore rendering them suitable to treat problems with very large number of classes using very reduced computational resources.
2. **Design strategies to optimize the ECOC coding designs**. Another objective of this dissertation is to exploit the distribution of the multi-class data to define problem-dependent ECOC designs that distribute their learning effort accordingly. In order to do so, we study different ways of optimizing ECOC coding matrices.
3. **Analyze and understand how error-correction is distributed among classes**. Error-correction has always been in the core of all ECOC analyses. Our objective in this dissertation is to show that the current way to analyze and understand error-correction for ECOC matrices does not reflect how the error-correction is distributed among classes. Hence, a novel way of representing ECOCs and their correcting capabilities has to be defined.

1.6 Contributions

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 the ECOC framework.

1. **Minimal ECOC Coding Design**: We propose to define the lower bound of an ECOC coding design in terms of the number of binary problems embedded in the coding design. We show that even with a very reduced number of binary classifiers we can obtain comparable results to state-of-the-art coding designs.
2. **Genetic Optimization of Minimal ECOCs**: Although showing a good performance, Minimal ECOCs are bounded in terms of generalization capabilities due to the reduced number of classifiers. In this sense, we propose to use a Genetic Algorithm to optimize the ECOC coding configuration and obtain a Minimal ECOC coding matrix with high generalization capabilities.
3. **ECOC-Compliant Genetic Algorithm**: Standard Genetic Algorithm use crossover and mutation operators that treat individuals as binary strings. This operators do not take into account the constraints of the ECOC framework and can lead to poor optimization schemes. We propose to redefine the standard crossover and mutation operators in order to take into account the constraints of the ECOC framework. In addition we also proposes and operator to dynamically adapt the code length of the ECOC during the training process. This redefinition of operators leads to very satisfying results in terms of classification performance as a function of the model complexity.

4. **ECOC separability and error-correction**: The error-correcting capability of an ECOC has always been depicted in literature as a single scalar, which hinders further analyses of error-correction between different categories. We propose to represent an ECOC by means of its separability matrix and use very simple heuristics to exploit the distribution of error-correction among pairs of classes to outperform state-of-the-art results.
5. **Error-Correcting Factorization**: Empowered by the novel representation of an ECOC as its Separability Matrix we propose to obtain an estimated Separability matrix from data using very simple statistics. Then, we defined the novel Error-Correcting Factorization to factorize that estimated Separability matrix into a discrete ECOC coding matrix that distributes error-correcting to those classes that are more prone to errors.
6. **Applications in Human Pose Recovery**: We applied the ECOC framework in the challenging problem of Human Pose Recovery, obtaining very satisfying results in comparison with state-of-the-art works. In addition we propose the HuPBA $8k+$ dataset, a large dataset for Human Pose Recovery in still images.

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

1.7 Thesis Outline

This dissertation is organized as follows:

In Part I, Chapter 2 introduces the basis of the Error-Correcting Output Codes (ECOCs), as well as the state-of-the-art coding and decoding designs. In addition, we also present good practices for ECOC coding and the constraints of ECOC coding matrices. In Chapter 3 we introduce a novel way to represent ECOCs by means of the separation between codewords and analyze how error-correcting capabilities are distributed for state-of-the-art designs.

Part II presents our contributions to problem-dependent coding designs for ECOCs using Genetic Algorithm in Chapter 4. In particular, we present the Minimal ECOC coding, which defines the lower bound in terms of the number of classifiers embedded in an ECOC coding matrix. Moreover, we also present a novel Genetic Algorithm that takes into account the constraints of the ECOC framework. Chapter 5 present a novel method to compute problem-dependent designs from ECOC based on the Error-Correcting Factorization, which factorizes an estimated codeword separability matrix obtained from training data.

In Part III Chapter 6 presents applications of ECOC codings in the challenging Computer Vision problem of Human Pose Estimation, in which we cast the problem of recognizing the human body parts as a classification problem at tackle it using the ECOC framework.

Finally, Part IV concludes the dissertation with a summary of contributions and future work in Chapter 7. Annex A provided technical details of datasets used for experimental results, and Annex B shows the complete list of publications generated as a result of this dissertation.

1.8 Notation

We introduce the notation for the this dissertation.

Bold capital letters denote matrices (e.g. $\mathbf{X}$), bold lower-case letters represent vectors (e.g., $\mathbf{x}$). All non-bold letters denote scalar variables. $\mathbf{x}^i$ is the $i-$th row of the matrix $\mathbf{X}$.

$\mathbf{x}_j$ is the $j-$th column of the matrix $\mathbf{X}$. $\mathbf{1}$ is a matrix or vector of all ones of the appropriate size. $\mathbf{I}$ is the identity matrix. diag($\mathbf{d}$) denotes a matrix with $\mathbf{d}$ in the main diagonal. x_{ij} denotes the scalar in the $i-$th row and $j-$th column of $\mathbf{X}$. $\|\mathbf{X}\|_F = \mathrm{tr}(\mathbf{X}^\top \mathbf{X})$ denotes the Frobenius norm. $\|\cdot\|_p$ is used to denote the Lp-norm. $\mathbf{x} \oplus y$ is an operator which concatenates vectors $\mathbf{x}$ and $\mathbf{y}$. rank($\mathbf{X}$) denotes the rank of $\mathbf{X}$. $\mathbf{X} \leq 0$ denotes the point-wise inequality.

Table 1.2 shows the different symbols used in this dissertation.

Matrix of training samples	$\mathbf{X}$	$i-$th training sample	$\mathbf{x}_i$
Vector of labels	$\mathbf{y}$	label of the $i-$th samples	y_i
i-th category	c^i	Indicator function	$\mathcal{I}$
ECOC coding matrix	$\mathbf{M}$	i-th codeword	$\mathbf{m}^i$
$i-$th dichotomy	$\mathbf{m}_i$	$i-$th classifier	f_i
Number of classes	k	Number of dichotomies	l
Vector of classifier predictions for the $i-$th sample	$\mathbf{f}(\mathbf{x}_i)$	Decoding measure	δ
Chromosome	$\mathbf{s}$	Error function	E
Number of Generations	G	$i-$th individual	I_i
Confusion matrix	$\mathbf{C}$	Performance of binary classifiers	$\mathbf{p}$
Dichotomy selection order	$\mathbf{t}$	Mutation control value	mt_c
Separability matrix	$\mathbf{H}$	i-th row of $\mathbf{H}$	$\mathbf{h}^i$
Design Matrix	$\mathbf{D}$	$i-$th row of $\mathbf{D}$	$\mathbf{d}^i$
Minimum distance between pairs of codewords	$\mathbf{P}$	$i-$th row of $\mathbf{P}$	$\mathbf{p}^i$
Matrix of eigenvectors	$\mathbf{V}$	Vector of eigenvalues	$\boldsymbol{\lambda}$
Likelihood map	$\mathbf{B}$	Weight matrix	$\mathbf{W}$

Table 1.2: Table of symbols.

Part I

Ensemble Learning and the ECOC framework

Chapter 2

2.1 The Error-Correcting Output Codes Framework

ECOC is a general multi-class framework built on the basis error-correcting principles in communication theory [39]. This framework suggests to view the task of supervised multi-class classification as a communication problem in which the label of a training example is being transmitted over a channel. The channel consists of the input features, the training examples, and the learning algorithm. Because of errors introduced by the finite number of training samples, choice of input features, and flaws in the learning process, the label information is corrupted. By coding the label information in an error-correcting codeword and transmitting (i.e learning) each bit separately (i.e., via uncorrelated independent binary classifiers), the algorithm may be able to recover from the errors produced by the number of samples, features or learning algorithm.

The ECOC framework is composed of two different steps: *coding* [3, 39] and *decoding* [47, 146]. At the coding step an ECOC coding matrix $\mathbf{M} \in \{-1,+1\}^{k\times l}$ is constructed, where k denotes the number of classes in the problem and l the number of bi-partitions defined to discriminate the k classes. In this matrix, the rows ($\mathbf{m}^i$'s, also known as error-correcting *codewords* or codewords for shorter) are univocally defined, since these are the identifiers of each category in the multi-class categorization problem. On the other hand, the columns of $\mathbf{M}$ denote the set of bi-partitions, dichotomies, or meta-classes to be learnt by each binary classifier f (also known as dichotomizer). In this sense, classifier f^j is responsible of learning the bi-partition denoted on the $j-$th column of $\mathbf{M}$. Therefore, each dichotomizer learns the classes with value $+1$ against the classes with value -1 in a certain column. Note that the ECOC framework is independent of the binary classifier applied. For notation purposes in further sections we will refer to the entry of $\mathbf{M}$ at the i-th row and the j-th column as $\mathbf{m}_{ij}$. Following this notation the i-th row (codeword of class c^i) will be referred as $\mathbf{m}^i$ and, the j-th column (j-th bi-partition or dichotomy) will be referred as $\mathbf{m}_j$.

Originally, the coding matrix was binary valued ($\mathbf{M} \in \{-1,+1\}^{k\times l}$). However, [3] introduced a third value, and thus, $\mathbf{M} \in \{-1,+1,0\}^{k\times l}$, defining ternary valued coding matrices. In this case, for a given dichotomy categories can be valued as $+1$ or -1 depending on the meta-class they belong to, or 0 if they are ignored by the dichotomizer. This new value allows the inclusion of well-known decomposition techniques into the ECOC framework, such has One vs. One [125] or Sparse [3] decompositions.

At the decoding step a sample $\mathbf{x}_t$ is classified among the k possible categories. In order to perform the classification task, each dichotomizer in f^j predicts a binary value for $\mathbf{x}_t$ whether it belongs to one of the bi-partitions defined by the correspondent dichotomy. Once the set of predictions $\mathbf{f}(\mathbf{x}_t) \in \mathbb{R}^l$ is obtained, it is compared to the codewords of $\mathbf{M}$ using a distance metric δ, known as the decoding function. The usual decoding techniques are based on well-known distance measures such as the Hamming or Euclidean distances. These measures were proven to be effective in binary valued ECOC matrices $\{+1, -1\}$. Nevertheless, it was not until the work of [47] that decoding functions took into account the meaning of the 0 value at the decoding step. Generally, the final prediction for $\mathbf{x}_t$ is given by the class c^i, where $\underset{i}{\text{argmin}}\ \delta(\mathbf{m}^i, \mathbf{f}(\mathbf{x}_t))$, $i \in \{1, \ldots, k\}$. Figure 2.1 shows an example of ECOC matrices for a toy problem.

By analysing the ECOC errors it has been demonstrated that ECOC corrects errors caused by the bias and the variance of the learning algorithm [81]. The variance reduction is to be expected, since ensemble techniques address this problem successfully [104] 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 $\mathbf{x}_t$ 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 [81] 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 [3].

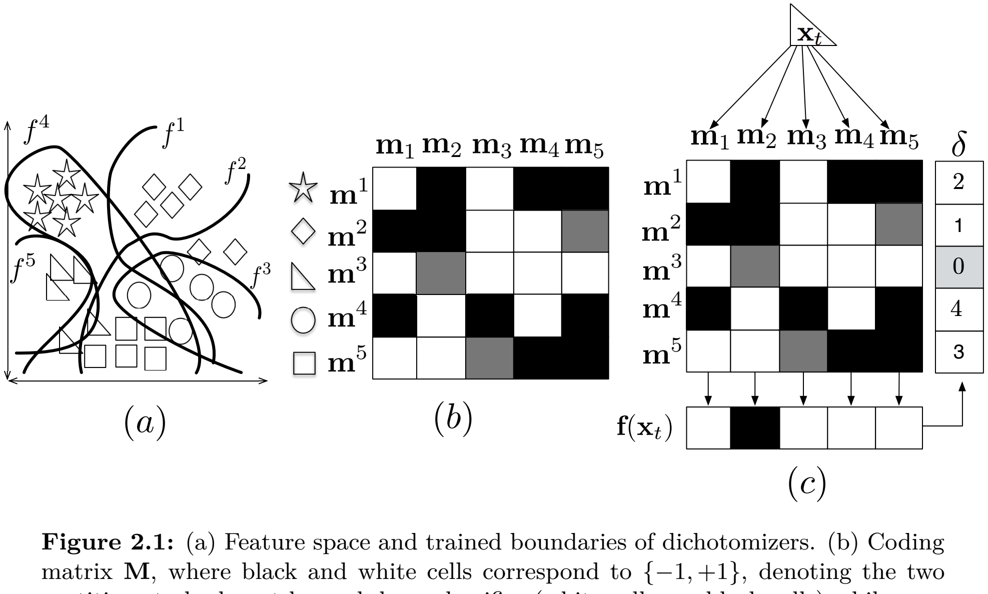


Figure 2.1: (a) Feature space and trained boundaries of dichotomizers. (b) Coding matrix $\mathbf{M}$, where black and white cells correspond to $\{-1, +1\}$, denoting the two partitions to be learnt by each base classifier (white cells vs. black cells) while grey cells correspond to 0 (ignored classes). (c) Decoding step, where the predictions of classifiers, $\{h^1, \ldots, h^5\}$ for sample $\mathbf{x}_t$ are compared to the codewords $\{\mathbf{m}^1, \ldots, \mathbf{m}^5\}$ and $\mathbf{x}_t$ is labelled as the class codeword at minimum distance.

2.2 ECOC Coding

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

2.2.1 Binary Coding

The standard binary coding designs are the One vs. All design [111] and the Dense Random design [3]. In One vs. All each dichotomy is defined to distinguish one category from the rest of the classes. An example is shown in Figure 2.2(a). The Dense Random coding generates a very high number of coding matrices $\mathbf{M}$ of length l, where the values $\{-1, +1\}$ are equiprobable. In [3] the authors suggested to experimentally set $l = 10 \log k$. From the set of Dense Random matrices generated the optimal one should maximize the minimum Hamming distance between the codewords of $\mathbf{M}$, taking into account that each column of $\mathbf{M}$ should contain $\{-1, +1\}$. An example of Dense Random coding is show in Figure 2.2(b). Furthermore, recent works have proposed various problem-dependent binary codings. In [141] the authors propose the Spectral Error-Correcting Output Codes [141], in which the ECOC coding corresponds to a subset of thresholded eigenvectors of the graph Laplacian of similarity matrix between categories.

2.2.2 Ternary Coding

The standard ternary codings designs are the One vs. One [125] and Sparse Random strategies [3]. The One vs. One strategy consider all possible pairs of classes, having a code length of $\frac{k(k-1)}{2}$. An example of the One vs. One coding design for a 4-class problem is show in Figure 2.2(c). The Sparse Random coding is similar to the Dense Random design, with the exception that it includes the 0 symbol. In this case, symbols $\{-1, +1, 0\}$ can not be equiprobable. In [3] the authors suggest an experimental length of $l = 15 \log k$ for the Dense Random design. In Figure 2.2(d) we show an example of the Sparse Random design. In addition, problem-dependent designs have also been the focus of recent works. In [106], the authors propose the Discriminant ECOC coding which 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. Finally, the work of Zhao et. al [142] proposes a dual projected gradient method embedded on a constrained concave-convex procedure to optimize an objective composed of a measure of expected problem separability, codeword correlation and regularization terms.

2.3 ECOC Decoding

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.

2.3.1 Binary Decoding

The binary decoding designs most frequently applied are: Hamming Decoding [100], Inverse Hamming Decoding [136], and Euclidean Decoding [3].

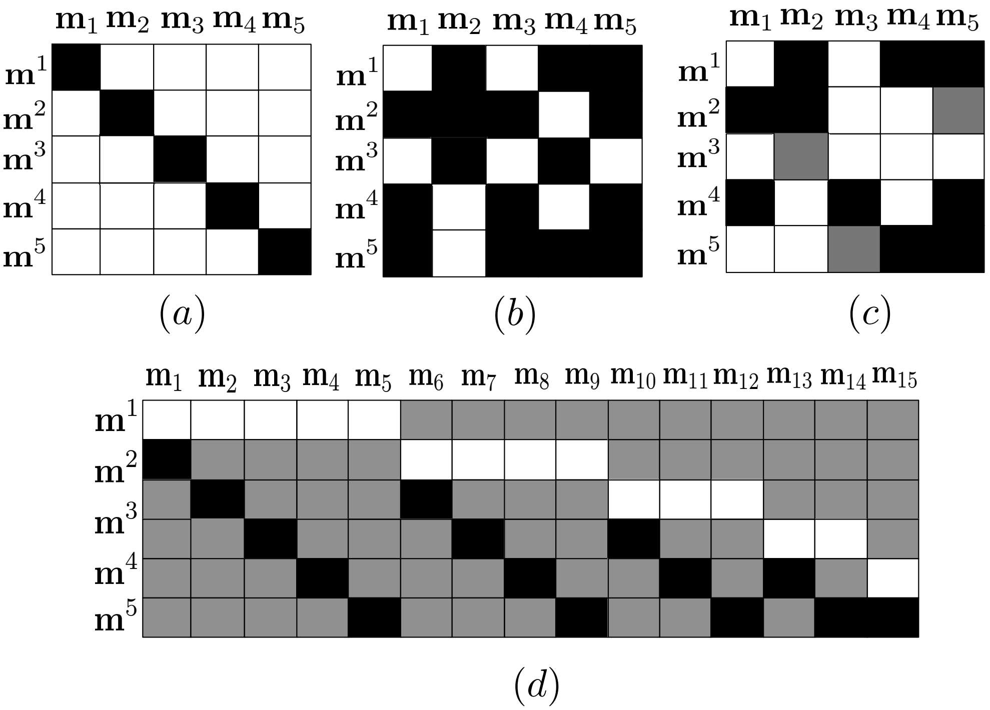

$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$
$\mathbf{m}^1$
$\mathbf{m}^2$
$\mathbf{m}^3$
$\mathbf{m}^4$
$\mathbf{m}^5$
(a)
(b)
(c)
$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$ $\mathbf{m}_6$ $\mathbf{m}_7$ $\mathbf{m}_8$ $\mathbf{m}_9$ $\mathbf{m}_{10}$ $\mathbf{m}_{11}$ $\mathbf{m}_{12}$ $\mathbf{m}_{13}$ $\mathbf{m}_{14}$ $\mathbf{m}_{15}$
(d)

Figure 2.2: (a) ECOC One vs. All coding design. (b) Dense Random coding desing. (c) Sparse Random coding. (d) One vs. One ECOC coding design.

- The **Hamming Decoding** is defined as follows:

$$\delta_{HD}(\mathbf{f}(\mathbf{x}_t), \mathbf{m}_i)) = \sum_{j=1}^{l} (1 - sign(h(\mathbf{x}_t)_j \cdot m_{ij}))/2 \qquad (2.1)$$

  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 [39].

- The **Inverse Hamming Decoding** introduced in [136] is defined as follows: let **H** be the matrix composed by the Hamming distance measures between the codewords of **M**. Each position of **H** is defined by $h_{ij} = \delta_{HD}(\mathbf{m}_i, \mathbf{m}_j) \forall i, j \in \{1, \ldots, k\}$. **H** can be inverted to find the vector containing the k individual class likelihood functions by means of:

$$\delta_{IHD}(\mathbf{f}(\mathbf{x}_t), \mathbf{m}_i) = \max(\mathbf{H}^{-1}\mathbf{r}^\top), \qquad (2.2)$$

  where the values of $\mathbf{H}^{-1}\mathbf{R}^\top$ can be seen as the proportionality of each class codeword in the test codeword, and $\mathbf{r}$ is the vector of Hamming Decoding values of the test codeword $\mathbf{f}(\mathbf{x}_t)$ for each of the base codewords $\mathbf{m}_i$. The practical behavior of the IHD showed to be very close to the behavior of the HD strategy [47].

- The **Euclidean Decoding** is another well-known decoding strategy based the Euclidean distance. This measure is defined as follows:

$$\delta_{ED}(\mathbf{f}(\mathbf{x}_t), \mathbf{m}_i) = \sqrt{\sum_{j=1}^{l} (h(\mathbf{x}_t)_j - m_{ij})^2} \tag{2.3}$$

2.3.2 Ternary Decoding

Concerning the ternary decoding, the state-of-the-art strategies are: Loss-based Decoding [47] and Probabilistic Decoding [47].

- The **Loss-based Decoding** strategy [47] chooses the label y_i that is most consistent with the predictions $\mathbf{h}$ (where $\mathbf{h}$ is a real-valued function $\mathbf{h} : \mathbf{X} \rightarrow \mathbf{y}$), in the sense that, if the data sample $\mathbf{x}_t$ was labeled y_t, the total loss on example $(\mathbf{x}_t, y_t)$ would be minimized over choices of $y_t \in \{1, \ldots, k\}$. Formally, given a Loss-function model, the decoding measure is the total loss on a proposed data sample $(\mathbf{x}_t, y_t)$:

$$LB(\mathbf{x}_t, \mathbf{m}_i) = \sum_{j=1}^{l} L(m_{ij} h(\mathbf{x}_t)_j), \tag{2.4}$$

  where $m_{ij} h(\mathbf{x}_t)_j$ 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 ξ_t according to the class i which codeword $\mathbf{m}_i$ obtains the minimum score.

- The **Probabilistic Decoding** strategy proposed in [102] is based on the continuous output of the classifier to deal with the ternary decoding. The decoding measure is given by:

$$\delta_{PD}(\mathbf{m}_i, F) = -\log\left(\prod_{j \in \{1,\ldots,l\} m_{ij} \neq 0} P(h(\mathbf{x}_t)_j = m_{ij} | h_j) + K \right), \tag{2.5}$$

  where K is a constant factor that collects the probability mass dispersed on the invalid codes, and the probability $P(h(\mathbf{x}_t)_j = m_{ij} | h_j)$ is estimated by means of:

$$P(h(\mathbf{x}_t)_j = m_{ij} | h_j) = \frac{1}{1 = e^{m_{ij}(\upsilon_j h_j + \omega_j)}}, \tag{2.6}$$

  where vectors υ and ω are obtained by solving an optimization problem [102].

2.4 Good practices in ECOC

Once the coding and decoding designs have been reviewed we analyze which are Good practices towards using the ECOC framework. Assume a k-class problem to be learned, then the ECOC framework will construct a matrix $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ in which k codewords will be chosen from the 3^l codes available. Following Newton's binomial this is be expressed as $\binom{3^l}{k}$ which denotes the huge size of the search space. A natural question that researchers have formulated is: What is a Good ECOC coding matrix? This question has been addressed in several works [4, 7, 39, 84, 141], each of them proposing different ways of evaluating an ECOC coding matrix, which are summed up in the following three properties:

1. **Correction capability**: the correction capability is expressed $\frac{\min(\delta(\mathbf{m}^i,\mathbf{m}^j))-1}{2}$, $\forall i,j \in \{1,\ldots,k\}$, $i \neq j$ [1]. In this sense, if $\min\delta(\mathbf{m}^i,\mathbf{m}^j)=3$, ECOC will be able to recover the correct multi-class prediction even if $\lfloor\frac{3-1}{2}\rfloor = 1$ binary classifier misses its prediction.
2. **Uncorrelated binary sub-problems**: the induced binary problems (columns of $\mathbf{M}$) should be as uncorrelated as possible for $\mathbf{M}$ to recover binary classifier errors.
3. **Use of powerful binary classifiers**: since the final class prediction consists of the aggregation of bit predictors, high-capacity binary classifiers are also required to obtain accurate multi-class predictions.
4. **Complexity of dichotomies**: easy binary problems will lead to increase overall performance of the ECOC coding [60, 142].

While exploiting these properties has lead to improvement either in overall model complexity [12, 60], scalability [142] or performance [51], we still lack an understanding about how the error-correction is distributed between different codewords and how that impacts the performance of the ECOC performance. This thesis aims to find answer to these issues.

2.5 ECOC constraints

To better understand ECOC correction we first start by analyzing the constraints of an ECOC coding matrix. Given the underlaying intuition of ECOCs in which an codeword $\mathbf{m}^i$ is assigned to each class, it follows that not all binary or ternary matrix $\mathbf{M} \in \{-1,+1,0\}$ are valid ECOC coding matrices. Hence, ECOC coding matrices have a certain structure and constraints that need to be taken into account. These constraints concern the repetitions of rows in $\mathbf{M}$. Since a repetition of rows will define two different categories with the same codeword, it implies that an ambiguous coding matrix $\mathbf{M}$ will be constructed. Moreover, error-correcting principles of communication theory are based on the assumption that errors introduced by each dichotomizer are uncorrelated [39]. In this sense, similar dichotomies will output correlated outputs, and thus, this situation has to be avoided. In the limit case, equivalent dichotomies will have equivalent outputs, and thus, no correction capability will be generated. In this sense, when designing ECOC coding matrices these constraints must be taken into account. We define a valid ECOC coding matrix $\mathbf{M} \in \{-1,+1,0\}^{k\times l}$ to be constrained by:

$$\min(\delta_{AHD}(\mathbf{m}^i,\mathbf{m}^k)) \geq 1,\ \forall i,j : i \neq j,\ i,j \in [1,\ldots,k], \tag{2.7}$$

$$\min(\delta_{HD}(\mathbf{m}_i,\mathbf{m}_j)) \geq 1,\ \forall i,j : i \neq j,\ i,j \in [1,\ldots,k], \tag{2.8}$$

$$\min(\delta_{HD}(\mathbf{m}_i,-\mathbf{m}_j)) \geq 1,\ \forall i,j : i \neq j,\ i,j \in [1,\ldots,k], \tag{2.9}$$

where δ_{AHD} and δ_{HD} are the Attenuated Hamming Distance (AHD) [47] and the Hamming Distance (HD), respectively, defined as follows:

$$\delta_{AHD}(\mathbf{m}^i,\mathbf{m}^j) = \sum_{k=1}^{n} |m_{ki}||m_{kj}|\mathcal{I}(m_{ki},m_{kj}), \tag{2.10}$$

[1] In the case of ternary codes this correction capability can be easily adapted.

$$\delta_{HD}(\mathbf{m}_i, \mathbf{m}_j) = \sum_{k=1}^{n} I(m_{ki}, m_{kj}),\ \ I(i,j) \begin{cases} 1 \text{ if } i \neq j \\ 0 \text{ otherwise.} \end{cases} \tag{2.11}$$

The motivation of using AHD to measure the distance between rows and HD to measure distance between columns is motivated by the different influences of the value 0 in columns and rows of $\mathbf{M}$. Thus, a position valued as 0 in a codeword means that a certain dichotomy is not taken into account in the definition of the class code, while for a dichotomy a position m_{ij} valued as 0 means that class c^j is ignored in the training step.

In this sense, by defining the ECOC constraints we have established the limits of the ECOC space. Intuitively, one may argue that inside that limited search space the ECOC matrix that yields the best validation performance is the optimal choice. However, this is unfeasible due to the large number of configurations allowed by the ECOC constraints. In addition, provided the discrete structure of the ECOC matrix, finding the optimal ECOC matrix given the dichotomizers has been demonstrated to be an NP-Complete problem [3]. While other works have exploited properties defined in previous sections, leading to several improvements, the basic representation of the code still remains intact, shedding no additional knowledge in the core property of ECOCs, which is the error-correction. We hypothesize that an alternative way of representing a coding matrix which encodes error-correcting structure is needed to better understand ECOCs, which is one of the main contributions of this thesis.

2.6 Conclusions

In this section, we have reviewed an analyze standard and state-of-the-art ECOC coding and decoding desings for binary and ternary ECOCs. We have also reviewed good practices when making use of the ECOC framework, providing fruitful insights on the lines of research the current works on problem-dependent ECOC techniques pursue. Finally, we have analyzed which are the constraints of ECOC matrices in terms of row and column distances, as well as, presenting the problem with current ECOC representations which do not provide a measure of how error-correction is distributed among class codewords.

Chapter 3

In this chapter we introduce the Separability matrix as a way to represent the error-correcting structure of an ECOC coding matrix. Although the concept of error-correction has always been in the core of all ECOC studies, up to this moment there has not been the need of defining explicitly a matrix of this kind. This is mainly due to the fact that predefined or random strategies assume that the coding matrix should have equidistant codewords or at least be as close as possible to an equidistant distribution [3]. The Separability matrix explicitly encodes the pairwise separation-distance between all pairs of codewords in $\mathbf{M}$.

3.1 The Separability matrix

One of the main properties of a good ECOC coding matrix is the correcting capability (the number of errors in binary classifiers that ECOC is able to correct). In literature, the correction capability of a coding matrix $\mathbf{M}$ is defined as $\frac{\min(\delta(\mathbf{m}^i,\mathbf{m}^j))-1}{2}$, $\forall i,j \in \{1,\ldots,k\}$, $i \neq j$. Therefore, distance between codewords and correction capability are directly related. Given this close relationship between distance and correction capability, we define the Separability matrix $\mathbf{H}$, as follows:

Given an ECOC coding matrix $\mathbf{M} \in \{-1,0,+1\}^{k\times l}$, the Separability matrix $\mathbf{H} \in \mathbb{R}^{k\times k}$ contains the distances between all pairs of codes in $\mathbf{M}$. Let $\{\mathbf{m}^i,\mathbf{m}^j\}$ be two codewords, the Separability matrix $\mathbf{H}$ at position (i,j), defined as h_{ij}, contains the Attenuated Hamming Distance between the codewords $\{\mathbf{m}^i,\mathbf{m}^j\}$, defined as $\delta_{AHD}(\mathbf{m}^i,\mathbf{m}^j)$. An example of Separability matrices for different coding designs is shown in Figure 3.1.

If we analyze the Separability matrix $\mathbf{H}$ of predefined ECOC coding designs [111, 125], we find that $\mathbf{H}$ is constant in all off-diagonal values. This means that codewords are equidistant, as shown in Figure 3.1(e)(h). In fact, when dealing with predefined codings, the Separability matrix makes little sense and has been overlooked since all off-diagonal values are constant. Nevertheless, in problem-dependent coding strategies the Separability matrix acquires a great value, since it shows which codewords are prone to have more errors due to the lack of error-correction capability.

3.2 From global to pair-wise correction capability

Correction capability has been a core objective of problem-dependent designs of $\mathbf{M}$. In previous works, different authors have always agreed on defining correction capability for an ECOC coding matrix as a scalar value [3, 39, 60, 62, 84, 141]. Hence, $\min(\mathbf{H})$ is expected to be large in order for $\mathbf{M}$ to recover from as many binary classifier errors as possible.

However, since $\mathbf{H}$ expresses the Attenuated Hamming distance between rows of $\mathbf{M}$, one can alternatively express the correction capability in a pair-wise fashion, allowing for a deeper understanding of how correction is distributed among codewords. Figure 3.2 shows an example of global and pair-wise correction capabilities calculation. Recall that the $\oplus$ operator between two vectors denotes its concatenation. Thus, we define the pair-wise correction capability as follows:

- The **pair-wise correction capability** of codewords $\mathbf{m}^i$ and $\mathbf{m}^j$ is expressed as: $\lfloor\frac{\min(\mathbf{h}^i \oplus \mathbf{h}^j)-1}{2}\rfloor$, where we only consider off-diagonal values of $\mathbf{H}$. This means that a sample of class c^i is correctly discriminated from class c^j even if $\lfloor\frac{\min(\mathbf{h}^i \oplus \mathbf{h}^j))-1}{2}\rfloor$ binary classifiers miss their predictions.

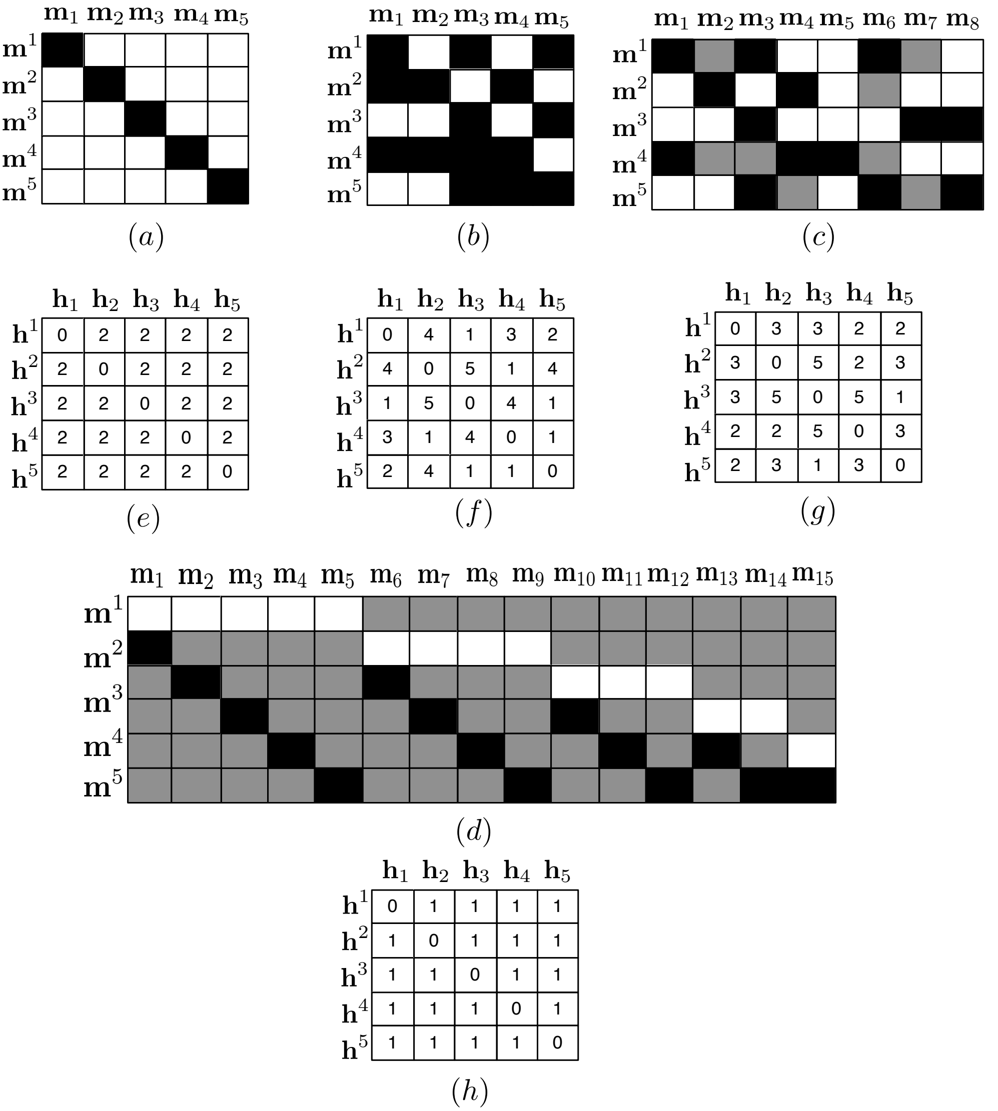
Note that though in Figure 3.2 the global correction capability of $\mathbf{M}$ is 0, there are pairs of codewords with a higher correction, e.g. $\mathbf{m}^2$ and $\mathbf{m}^8$. In this case the global correction capability as defined in literature is overlooking ECOC coding characteristics that can potentially be exploited. This novel way of expressing the correction capability of an ECOC matrix enables a better understanding of how ECOC coding matrices distribute their correction capability, and gives an insight on how to design coding matrices.

3.3 Conclusions

In this chapter we have show that the standard way of analyzing the error-correction of an ECOC clearly overlooks crucial aspect in the pair-wise relationships between codewords that can be beneficial to exploit. We have also shown how to compute the Separability Matrix $\mathbf{H}$ given a ECOC coding matrix $\mathbf{M}$ and what structure is expected for state-of-the-art coding designs. However, we still need to solve various issues to exploit this representation:

- How to estimate a suitable Separability matrix for a multi-class problem given only the multi-class data $\{\mathbf{x}_i, y_i\} : i \in \{1, \ldots, n\}$, which we denote as the Design Matrix $\mathbf{D} \in \mathbb{R}^{k \times k}$. This problem intuitively tackles how error-correction should be allocated. The open question is of the following form: Should correction be allocated according to those classes that are more prone to error, in order for them to have better recovery behavior (i.e. following a *"no class is left behind"* criteria)? Or should correction be allocated to easily separable classes [60, 141, 142] (i.e. following a *"hard classes are left behind"* scheme)?
- Given the Design Matrix $\mathbf{D}$ how to obtain the ECOC matrix $\mathbf{M}$.

Both of this issues are analyzed in Section 5.1.


$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$
$\mathbf{m}^1$ $\mathbf{m}^2$ $\mathbf{m}^3$ $\mathbf{m}^4$ $\mathbf{m}^5$
(a)
$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$
$\mathbf{m}^1$ $\mathbf{m}^2$ $\mathbf{m}^3$ $\mathbf{m}^4$ $\mathbf{m}^5$
(b)
$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$ $\mathbf{m}_6$ $\mathbf{m}_7$ $\mathbf{m}_8$
$\mathbf{m}^1$ $\mathbf{m}^2$ $\mathbf{m}^3$ $\mathbf{m}^4$ $\mathbf{m}^5$
(c)
$\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$ $\mathbf{m}_6$ $\mathbf{m}_7$ $\mathbf{m}_8$ $\mathbf{m}_9$ $\mathbf{m}_{10}$ $\mathbf{m}_{11}$ $\mathbf{m}_{12}$ $\mathbf{m}_{13}$ $\mathbf{m}_{14}$ $\mathbf{m}_{15}$
$\mathbf{m}^1$ $\mathbf{m}^2$ $\mathbf{m}^3$ $\mathbf{m}^4$ $\mathbf{m}^5$
(d)

	$\mathbf{h}_1$	$\mathbf{h}_2$	$\mathbf{h}_3$	$\mathbf{h}_4$	$\mathbf{h}_5$
$\mathbf{h}^1$	0	2	2	2	2
$\mathbf{h}^2$	2	0	2	2	2
$\mathbf{h}^3$	2	2	0	2	2
$\mathbf{h}^4$	2	2	2	0	2
$\mathbf{h}^5$	2	2	2	2	0

(e)

	$\mathbf{h}_1$	$\mathbf{h}_2$	$\mathbf{h}_3$	$\mathbf{h}_4$	$\mathbf{h}_5$
$\mathbf{h}^1$	0	4	1	3	2
$\mathbf{h}^2$	4	0	5	1	4
$\mathbf{h}^3$	1	5	0	4	1
$\mathbf{h}^4$	3	1	4	0	1
$\mathbf{h}^5$	2	4	1	1	0

(f)

	$\mathbf{h}_1$	$\mathbf{h}_2$	$\mathbf{h}_3$	$\mathbf{h}_4$	$\mathbf{h}_5$
$\mathbf{h}^1$	0	3	3	2	2
$\mathbf{h}^2$	3	0	5	2	3
$\mathbf{h}^3$	3	5	0	5	1
$\mathbf{h}^4$	2	2	5	0	3
$\mathbf{h}^5$	2	3	1	3	0

(g)

	$\mathbf{h}_1$	$\mathbf{h}_2$	$\mathbf{h}_3$	$\mathbf{h}_4$	$\mathbf{h}_5$
$\mathbf{h}^1$	0	1	1	1	1
$\mathbf{h}^2$	1	0	1	1	1
$\mathbf{h}^3$	1	1	0	1	1
$\mathbf{h}^4$	1	1	1	0	1
$\mathbf{h}^5$	1	1	1	1	0

(h)

Figure 3.1: ECOC codings of One vs. All (a), Dense Random (b), Sparse Random (c) and One vs. One (d). Separability matrices for One vs. All (e), Dense Random (f), Sparse Random (g) and One vs. One coding (h). Note how predefined coding designs generate constant separability matrices, while other designs present different distributions of separation between codewords.

M

	■	■	■	■	■	□	□
$\mathbf{m}^2$	■	■	□	□	□	□	■
	■	□	■	■	■	□	□
	■	□	□	■	□	□	□
	□	■	■	■	■	□	□
	□	■	□	□	■	□	□
	□	□	■	■	■	□	□
$\mathbf{m}^8$	□	□	□	□	□	■	□

$$\lfloor \frac{\min(\mathbf{H}) - 1}{2} \rfloor = 0$$

H

	0	4	1	3	1	3	2	6
$\mathbf{h}^2$	4	**0**	5	3	5	3	6	4
	1	5	**0**	2	2	4	1	5
	3	3	2	**0**	4	4	3	3
	1	5	2	4	**0**	2	1	5
	3	3	4	4	2	**0**	3	3
	2	6	1	3	1	3	**0**	4
$\mathbf{h}^8$	6	4	5	3	5	3	4	**0**

$$\lfloor \frac{\min(\mathbf{h}^2 \oplus \mathbf{h}^8) - 1}{2} \rfloor = 1$$

Figure 3.2: Example of global versus pair-wise correction capability. On the left side of the figure the calculation of the global correction capability is shown. The right side of the image shows a sample of pair-wise correction calculation for codewords $\mathbf{m}^2$ and $\mathbf{m}^8$.

Part II

Learning ECOC representations

Chapter 4

Learning ECOCs using Genetic Algorithms

Due to the nature of the Error-Correcting Output Codes space [62, 89, 90] Genetic Algorithms (GA) have been applied in this scenario to learn problem-dependent ECOC coding matrices. In particular, the ECOC search space has been proven to be non-continuous and non-differentiable, thus making the use of Genetic Algorithms a popular choice . In this chapter we present our contributions to state-of-the-art Genetic Algorithms for ECOCs, developing methods based both on standard and non standard GAs that are compliant with the Error-Correcting Output Codes constraints.

4.1 Towards Reducing the ECOC Code Length

The coding step of the ECOC framework has been widely studied in literature, obtaining either predefined [111, 125], random [3] or problem-dependent [49, 105, 106, 122, 145] coding designs. The most well-known coding schemes are the predefined ones, such as, One vs. One [125] and One vs. All [111] designs, in which $\binom{k}{2}$ and k dichotomies are defined, respectively. In the One vs. One scheme all the possible pair-wise groups of the k categories are defined, while in the One vs. All scheme each dichotomy is responsible of discriminating one class against the rest of the classes. In contra-position, some works in literature have stated that random designs [4], with code length of $\{10, 15\} \cdot \log_2(k)$ can perform as well as predefined codes. However, predefined or random designs do not exploit the problem-domain information.

In this sense, some works in literature have proposed the use of problem-dependent ECOC coding matrices. Problem-dependent coding designs lay on the assumption that predefined and random codes may not be suitable to solve a given problem since they do not take into account the underlying distribution of the classes in the multi-class problem. In [106] the authors proposed the DECOC coding design of length $k - 1$ in which a tree structure is embedded in the ECOC coding matrix, where nodes correspond to classes that maximize a split criterion. In the trend of the previous works, [49] proposed the same tree embedding where the nodes correspond to the most difficult meta-classes to be learnt. Other works aim to treat the problem either by soft weight sharing methods [5] or by using EM algorithm to find the optimal decomposition of the multi-class problem [128]. However, in [111] Rifkin et. al stated that when using high capacity dichotomizers the code length can be reduced with no loss of generalization, and tested their hypothesis in the One vs. All coding design using k dichotomies. Nevertheless, few are the works that aim to reduce the code length by using problem-dependent designs [106].

In general, classification performance has always been the core of all ECOC evaluation, regardless of its length. Nevertheless, following the Occam's razor principle, in equal conditions, simpler models tend to be more suitable. In this sense, we can consider that in the ECOC framework the number of classifiers has a direct relationship to the complexity of the model. For instance, when using SVMs as the base classifier, the number of classifiers has a direct relationship to the overall number of Support Vectors (SVs) of the ECOC coding design. At the same time, the number of SVs is directly proportional to the complexity in the ECOC decoding step. Thus, a trade off between generalization performance and code length has to be taken into account in order to perform a fair analysis of ECOC capabilities. In Figure 4.1 we show the number of classifiers defined for some of the state-of-the-art coding designs with respect to the number of classes of the multi-class problem. The coding designs taken into account are the Minimal ECOC, Dense Random, Sparse Random, DECOC, One vs. All, ECOC-ONE, Forest-ECOC, and One vs. One [4, 10, 48, 49, 106, 111, 125].

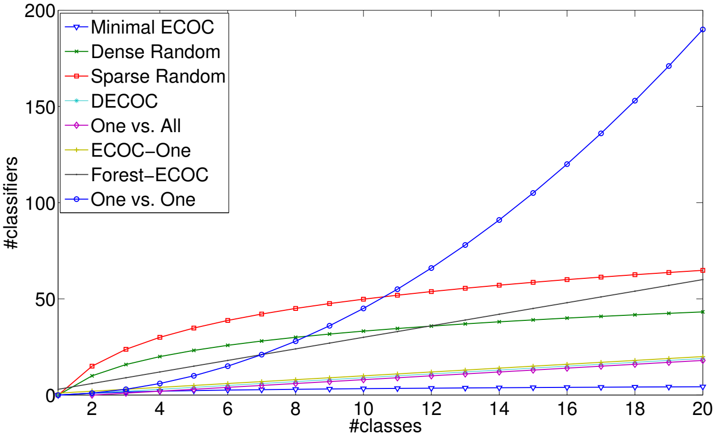


Figure 4.1: Number of classifiers per coding design and number of classes.

Note the great difference between the number of dichotomies defined by state-of-the-art strategies. In this case we can see that the Minimal ECOC approach [10] defines the most reduced code length in contra-position with the One vs. All and One vs. One strategies, which have a linear and quadratic growth with the number of classes, respectively. This fact encourages the use of sub-linear ECOC strategies (with respect to the number of classifiers used), since the scalability problem that is present when using other strategies can be easier to tackle. In the next section, we pose the process for designing Minimal ECOCs.

4.2 Learning Minimal ECOCs using standard Genetic Algorithms

4.2.1 Minimal ECOC Coding

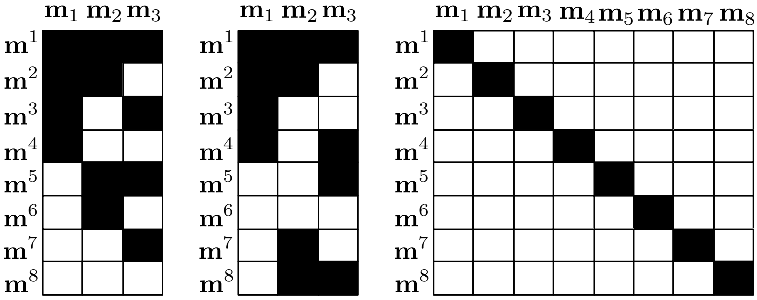
The use of large codewords has been suggested thoroughly in ECOC literature [39, 49, 125], in order to correct as many errors as possible at the decoding step. However, if high effort is put into improving the robustness of each individual dichotomizer very reduced codewords can be defined in order to save time. A clear example is the One vs. All ECOC coding, which has been widely applied in a large number of problems, and it has recently gained a lot of attention in the deep-learning large-scale setting [67, 73, 124].

Although One vs. All, DECOC, Dense, and Sparse Random approaches have a relatively small codeword length, we can take advantage of the information theory principles to obtain a Minimal definition of the ECOC coding matrix. Having a k-class problem, the minimum number of bits necessary to codify and univocally distinguish k codes is:

$$l = \lceil \log_2 k \rceil \tag{4.1}$$

where $\lceil . \rceil$ rounds to the upper integer.

For instance, we can think of a codification where the class codewords correspond to the k first Gray or binary code sequences of l bits, defining the Gray or binary Minimal ECOC designs. Note that this design represents the Minimal ECOC codification in terms of the codeword length. An example of a binary Minimal ECOC, Gray Minimal ECOC, and One vs. All ECOC designs for a 8-class problem are shown in Figure 4.2. The white and black positions correspond to the symbols $+1$ and -1, respectively. The reduced number of classifiers required by this design in comparison with the state-of-the-art approaches is shown in the graphic of Figure 4.1.



(a) (b) (c)

Figure 4.2: (a) Binary Minimal, (b) Gray Minimal, and (c) One vs. All ECOC coding designs of a 8-class problem.

Besides exploring predefined binary or Gray Minimal coding matrices, we also propose the design of a different Minimal codification of M based on the distribution of the data and the characteristics of the applied base classifier, which can increase the discrimination capability of the system. However, finding a suitable Minimal ECOC matrix for an $k-$class problem requires to explore all the possible $\frac{2^l!}{2(2^l-k)!}$ binary matrices, where l is the minimum codeword length in order to define a valid ECOC matrix. For this reason, we propose an evolutionary parametrization of the Minimal ECOC design.

4.2.2 Evolutionary Minimal ECOC Learning

When defining a Minimal design of an ECOC, the possible loss of generalization performance due the reduced number of dichotomies has to be taken into account. In order to deal with this problem an evolutionary optimization process is used to find a Minimal ECOC with high generalization capability.

In order to show the parametrization complexity of the Minimal ECOC design, we first provide an estimation of the number of different possible ECOC matrices that we can build, and therefore, the search space cardinality. We approximate this number using some simple combinatorial principles. First of all, if we have an $k-$class problem and l possible bits to represent all the classes, we have a set with 2^l different codewords. In order to build an ECOC matrix, we select k codewords from the set without reposition. In other words, taking k codewords from a variation of 2^l. In combinatorics this number is represented as $\binom{k}{2^l}$, which means that we can construct $\frac{2^l!}{(2^l-k)!}$ different ECOC matrices. Nevertheless, in the ECOC framework, one matrix and its opposite (swapping all minus ones by ones and vice-versa) are considered as the same matrix, since both represent the same partitions of the data. Therefore, the approximated number of possible ECOC matrices with the minimum number of classifiers is $\frac{2^l!}{2(2^l-k)!}$. In addition to the huge cardinality, it is easy to show that this space is non continuous neither differentiable, because a change in just one bit of the matrix may produce a wrong coding design (i.e one in which two rows of $\mathbf{M}$ are equivalent). In [34] the authors proved that finding the optimal codeword of the matrix $\mathbf{M}$ is computationally unfeasible since it is an *NP-complete* problem.

In this type of scenarios evolutionary approaches are often introduced with good results. Evolutionary algorithms are a wide family of methods that are inspired on Darwin's evolution theory, and are formulated as optimization processes where the solution space is not differentiable or is not well defined, hence this kind of approaches can only guarantee to find suboptimal solutions in the best case.

In these cases, the simulation of natural evolution processes using computers results in stochastic optimization techniques which often outperform classical methods of optimization when applied to difficult real-world problems. Although the most used and studied evolutionary algorithms are the Genetic Algorithms, the *Population Based Incremental Learning* (PBIL) by Baluja and Caruana [9] proposed a new family of evolutionary methods which is striving to find a place in this field. In contrast to GAs, this new algorithms consider each value in the chromosome as a random variable, and their goal is to learn a probability model to describe the characteristics of good individuals. In the case of PBIL, if a binary chromosome is used, a uniform distribution is learnt in order to estimate the probability of each variable to be one or zero. In this thesis we report experiments made with both GA and PBIL evolutionary strategies.

Problem encoding

The first step in order to use an evolutionary algorithm is to define the problem encoding, which consists of the representation of a certain solution or point in the search space by means of a *genotype* or alternatively a *chromosome* [70]. When the solutions or individuals are transformed in order to be represented in a chromosome, the original values (the individuals) are referred as *phenotypes*, and each one of the possible settings for a phenotype is the *allele*. Binary encoding is the most common, mainly because first works about GA used this type of encoding. In binary encoding, every chromosome is a string of bits. Although this encoding is often not natural for many problems and sometimes corrections must be performed after crossover and/or mutation, in our case, the chromosomes represent binary ECOC matrices, and therefore, this encoding adapts to the problem. Each ECOC is encoded as a binary chromosome $\mathbf{s} = \{m_{11}, m_{12}, \ldots, m_{kl}\}$, where $m_{ij} \in \{-1, +1\}$ corresponds to the $i - th$ column of the $j - th$ row in the coding matrix $\mathbf{M}$.

Adaptation function

Once the encoding is defined, we need to define the adaptation function, which associates to each individual its adaptation value to the environment, and thus, their survival probability. In the case of the ECOC framework, the adaptation value must be related to the classification error.

Given a chromosome $\mathbf{s} = \{s_0, s_1, \ldots, s_{k \times l}\}$ with $s_i \in \{0, 1\}$ (equivalent to $s_i \in \{-1, +1\}$), the first step is to recover the ECOC matrix $\mathbf{M}$ codified in this chromosome. The values of $\mathbf{M}$ allows to create binary classification problems from the original multi-class problem, denoted by the partitions defined by the columns of M. Each binary problem is addressed by means of a binary classifier, which is trained in order to separate both partitions of classes. Once the classifiers are trained for each binary problem in a subset of the training data, the adaptation value corresponds to the multi-class classification error of the ECOC design.

The adaptation value for an individual represented by a certain chromosome $\mathbf{s}_i$ can be formulated as:

$$E(\mathbf{X}, \mathbf{y}, \mathbf{M}_i) = \frac{\sum_{j=1}^{j'} \mathcal{I}(\delta(\mathbf{M}_i, \mathbf{f}(\mathbf{x}_j)), y_j)}{j'} \tag{4.2}$$

where $\mathbf{M}_i$ is the ECOC matrix encoded in $\mathbf{s}_i$, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{tr}\} \in \mathbb{R}^{1 \times f}$ a set of samples, $\mathbf{y} = \{y_1, \ldots, y_{tr}\} \in [1, \ldots, k]$ the labels for samples in $\mathbf{X}$. In addition, δ is the function that returns the multi-class label applying the ECOC decoding strategy, taking as input parameters the output vector of trained classifiers on sample, denoted as $\mathbf{f}(\mathbf{x}_j)$ and the coding matrix $\mathbf{M}_i$. Finally $\mathcal{I}$ denotes the indicator function.

Evolutionary process

Once the encoding and adaptation function have been defined, we use standard implementations for GA and PBIL, in order to evolve the Minimal ECOC matrices. In the case of GA, the scattered crossover operator is used, in which we generate a random binary vector with a binary value assigned to each gene. The first child is created using all the genes from the first parent in those positions with a value of one, and the genes of the second parent with positions with the value zero. The second child is created as the complementary of the first one. That is, taking genes from the second parent for values one and from the first parent for values zero. In order to introduce variations to the individuals, we use mutation operator that adds a unit Gaussian distributed random value to the chosen gene. In the genetic optimization process the chromosome coding a certain ECOC Minimal matrix $\mathbf{M}$ is changed within generations due to crossover or mutation transformations. In this sense, there is no warranty that after any transformations the new chromosome will represent a valid ECOC matrix. This issue has been treated defining bounds for every gene, which contemplate the possible non-ECOC-compliant transformations. In this scope, if a chromosome represents a non-ECOC-compliant matrix then its fitness value (classification error) will be set to the upper bound, which is 1. For PBIL, the best two individuals of each population are used to update the probability distribution. At each generation, this probability distribution is used to sample a new population of individuals. An uniform random noise is applied to the probability model to avoid convergence to local minima.

Finally, in both evolutionary strategies we adopt an *Island Model* evolution scheme in order to exploit a more coarse grain parallel model. The main idea is to split a population of individuals into sub-populations. If each sub-population is evolved independently from the others, genetic drift will tend to drive these populations in different directions. By introducing migration, the *Island Model* is able to exploit differences in the various sub-populations (this variation in fact represents a source of genetic diversity). Each sub-population is an island and there is a chance movement of genetic material from one island to another.

Training the binary classifiers

In [111], Rifkin concludes that the number of classifiers in the ECOC problem can be successfully reduced using high capacity classifiers. In this scope, in [72] SVMs with a RBF kernel have demonstrated to have $VC - dimension = \infty$. Therefore, in this paper we adopt the Support Vector Machines with Gaussian Radial Basis Functions kernel (SVM-RBF) as our base classifier. Training a SVM implies the selection of a subset of data points (the support vectors), which are used in order to build the classification boundaries. In the specific case of SVMs with Gaussian RBF kernels, we need to optimize the kernel parameter $\gamma = \frac{1}{\sigma^2}$ and the regularizer C, which have a close relation to the data distribution. While the support vectors selection is part of the SVM learning algorithm, and therefore, is clearly defined, finding the best C and γ is addressed in literature with various heuristic or brute-force approaches. The most common approach is the use of cross-validation processes which select the best pair of parameters for a discretization of the parameters space. Nevertheless, this can be viewed as another optimization problem and it can be faced using evolutionary algorithms. For each binary problem, defined by one column of the ECOC matrix, we use Genetic Algorithms in order to find good values for C and γ, using the same settings than in [90], where individuals correspond to a pairs of genes, and each gene corresponds to the binary codification of a floating point value. As for any model selection procedure, this parameter estimation is performed under a 2-fold cross-validation measurement to reduce the overfitting bias. More details can be found in [80].

Figure 4.3 illustrates an iteration of the evolutionary Minimal ECOC parametrization for a toy problem of three categories. Given an input chromosome $\mathbf{s}_1$ that codifies a valid Minimal matrix $\mathbf{M}$, for each dichotomizer, an evolutionary approximation of the classifier parameters $\{C, \gamma\}$ is performed. From left to right of the image, we show the Minimal ECOC matrix codifying the 3-class problem, the feature spaces, and the search space of $\{C, \gamma\}$. Each decision boundary shows a possible certain solution given by the GA. In this sense, the GA starts giving solutions (pairs $\{C, \gamma\}$) that do not fit a robust decision boundary at the beginning, but after convergence, the GA increases the adjustment of these boundaries. The final parameters are those that minimize the error over a validation subset. The evolution of the best individual is shown in the toy error surface on the right of the image. Finally, the evolution of this matrix returns the adaptation value E of the current individual (current Minimal configuration), estimated as the minimum classification error computed.

This coding optimization process is computationally more expensive than the standard approaches since for every Minimal coding matrix $\mathbf{M}$ all the binary classifiers $\{h^1 \dots h^l\}$ have to be optimized. Nevertheless, large-scale multi-classification problems which are typically computationally unfeasible using standard coding designs can be treated with this approach, since this optimization is only applied for a reduced number of dichotomizers. Furthermore, we use as speed up hack by storing a cache of trained classifiers. Consider a random iteration of the optimization. In this iteration, every Minimal coding matrix will define a set of bi-partitions $\{\mathbf{m}_1, \dots, \mathbf{m}_l\}$ to be learnt by a set of dichotomizers $\{h^1 \dots h^l\}$. In fact, we can assume that a certain bi-partition $\mathbf{m}_i$ learnt by a certain dichotomizer h^i will be

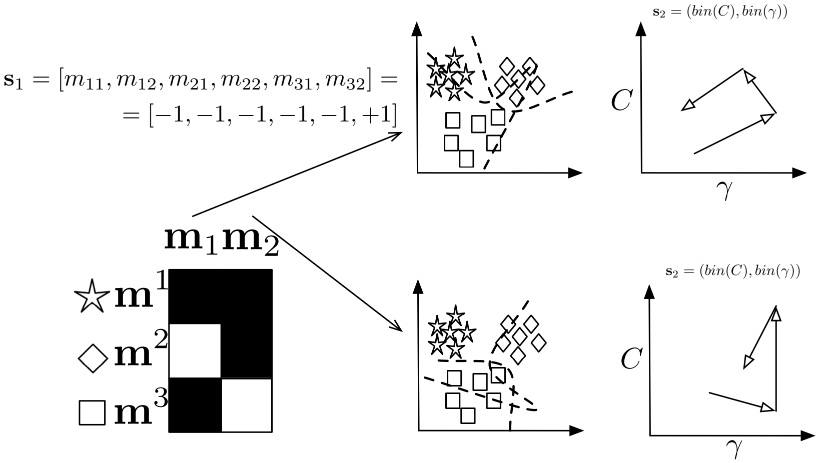

$s_1 = [m_{11}, m_{12}, m_{21}, m_{22}, m_{31}, m_{32}] =$
$= [-1, -1, -1, -1, -1, +1]$
$\mathbf{m}_1\mathbf{m}_2$
☆$\mathbf{m}^1$
◇$\mathbf{m}^2$
□$\mathbf{m}^3$
$s_2 = (bin(C), bin(\gamma))$
C
γ
$s_2 = (bin(C), bin(\gamma))$
C
γ

Figure 4.3: Nested evolutionary optimization process for 3-class toy problem.

repeated among the Minimal coding matrices along generations because of the nature of the evolutionary optimization process. Hence in order to save time and not retrain classifiers a cache of column codification and parameter optimization is saved during the evolutionary parametrization process.

4.2.3 Experiments

In order to present the results, first, we discuss the data, methods, and evaluation measurements of the experiments.

Data

The data used on these experiments are the following UCI datasets [8]: *Demathology*, *Iris*, *Ecoli*, *Vehicle*, *Wine*, *Segmentation*, *Glass*, *Thyroid*, *Vowel*, *Balance*, *Shuttle*, and *Yeast*.

In addition we performed experiments in five challenging Computer Vision datasets: *Labeled Faces In The Wild*, *Traffic Sign*, *ARface*, *Cleafs and Accidental*, and *MPEG7*. Detailed descriptions and characteristics for each dataset can be found in Appendix A.

Methods

We compare the One vs. One [125] and One vs. All [103] ECOC, DECOC [106] and Forest-ECOC [49]. approaches with the binary and evolutionary Minimal approaches (Genetic and PBIL [9]). For simplicity we omitted the Gray Minimal design. The Hamming decoding is applied at the decoding step [39]. The ECOC base classifier is the libsvm implementation of SVM with Radial Basis Function kernel [101]. The SVM C and γ parameters for each binary classifier are tuned via Genetic Algorithms and PBIL for all the methods, minimizing the classification error of a two-fold evaluation over the training sub-set.

• *Evaluation measurements*: The classification performance is obtained by means of a stratified ten-fold cross-validation, and tested for the confidence interval with a two-tailed t-test. We also apply the Friedman and Nemenyi tests [38] in order to look for statistical significance among the obtained performances.

Experimental classification results

The classification results obtained for all the datasets considering the different ECOC configurations are shown in Table 4.1. In order to compare the performances provided for each strategy, the table also shows the mean rank of each ECOC design considering the 17 different experiments[1]. The rankings are obtained estimating each particular ranking r_{ij} for each problem i and each ECOC configuration j, and computing the mean ranking $\overline{\mathbf{r}_j}$ for each design as $\overline{\mathbf{r}_j} = \frac{1}{N}\sum_i r_{ij}$, where N is the total number of datasets. We also show the mean number of classifiers (#) required for each strategy.

Table 4.1: Classification results and number of classifiers per coding design.

	Binary Minimal ECOC		GA Minimal ECOC		PBIL Minimal ECOC		one-vs-all ECOC	
Dataset	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.
Derma	96.0±2.9	3	96.3±2.1	3	95.1±3.3	3	95.1±5.1	6
Iris	96.4±6.3	2	**98.2±1.9**	2	96.9±6.0	2	96.9±3.0	3
Ecoli	80.5±10.9	3	**81.4±10.8**	3	79.5±12.2	3	79.5±13.8	28
Vehicle	72.5±14.3	2	76.99±12.4	2	74.2±13.4	3	74.2±9.7	4
Wine	95.5±4.3	2	97.2±2.3	2	95.5±4.3	2	95.5±2.1	3
Segment	96.6±2.3	3	96.6±1.5	3	96.1±1.8	3	96.1±2.4	7
Glass	56.7±23.5	3	50.0±29.7	3	53.85±25.8	3	53.8±26.2	7
Thyroid	96.4±5.3	2	93.8±5.1	2	95.6±7.4	2	95.6±5.4	3
Vowel	57.7±29.4	3	**81.78±11.1**	3	80.7±11.9	3	80.7±15.2	11
Balance	80.9±11.2	2	87.1±9.2	2	89.9±8.4	2	89.9±6.1	3
Shuttle	80.9±29.1	3	83.4±15.9	3	**90.6±11.3**	3	90.6±18.1	7
Yeast	50.2±18.2	4	54.7±11.8	4	51.1±18.0	4	51.1±18.3	10
FacesWild	26.4±2.1	10	**30.7±2.3**	10	30.0±2.4	10	25.0±3.9	184
Traffic	90.8±4.1	6	90.6±3.4	6	90.7±3.7	6	91.8±4.6	36
ARFaces	76.0±7.2	6	85.84.0±5.2	6	84.2±5.3	6	84.0±6.3	50
Clefs	81.2±4.2	3	81.8±9.3	3	81.7±8.2	3	80.8±11.2	7
MPEG7	89.29±5.1	7	90.4±4.5	7	90.1±4.9	7	87.8±6.4	70
Rank & #	5.2	**3.7**	3.6	**3.7**	3.0	**3.7**	4.8	22.8

	one-vs-one ECOC		DECOC		Forest-ECOC	
Dataset	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.
Derma	94.7±4.3	15	**97.1±3.3**	5	96.0±2.8	15
Iris	96.3±3.1	3	97.6±5.4	2	96.9±4.1	6
Ecoli	79.2±13.8	28	69.4±10.3	7	75.2±9.5	21
Vehicle	83.6±10.5	6	**84.1±13.3**	2	81.6±15.3	9
Wine	97.2±2.4	3	97.7±2.4	2	**97.8±2.3**	6
Segment	**97.2±1.3**	21	97.0±1.4	6	97.1±2.1	18
Glass	**60.5±26.9**	15	55.1±28.5	6	46.9±29.1	15
Thyroid	96.1±5.4	3	96.0±5.1	2	**97.6±4.3**	6
Vowel	78.9±14.2	28	66.7±13.3	10	70.1±12.3	30
Balance	**92.8±6.4**	3	82.8±8.3	2	92.2±8.7	6
Shuttle	86.3±18.1	21	77.1±17.4	6	84.3±14.5	18
Yeast	52.4±20.8	45	55.8±21.2	9	**56.0±17.6**	27
FacesWild	-	16836	26.3±3.1	183	29.1±2.5	471
Traffic	90.6±4.1	630	86.2±4.2	35	**96.7±3.5**	105
ARFaces	**96.0±2.5**	190	82.7±49	9	85.6±0	147
Clefs	84.2±6.8	21	96.9±6.4	6	**97.1±5.3**	18
MPEG7	**92.8±3.7**	2415	83.4±4.5	69	88.9±5.3	207
Rank & #	3.7	1193.1	4.2	21.8	3.1	66.2

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 (using notation as in the seminal paper of Demsar et al. [38] for this section):

$$X_F^2 = \frac{17N}{k(k+1)}\left[\sum_j R_j^2 - \frac{(k+1)^2}{4}\right] \tag{4.3}$$

[1] One vs. One coding for LFW dataset has been omitted for being computationally unfeasible

In our case, with $k = 7$ methods to compare, $X_F^2 = -9.92$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{4.4}$$

Applying this correction we obtain $F_F = 1.72$. With seven methods and seventeen experiments, F_F is distributed according to the F distribution with 42 and 272 degrees of freedom. The critical value of $F(42, 272)$ for 0.05 is 1.45. As the value of F_F is higher than 1.45 we can reject the null hypothesis.

Once we have checked for the non-randomness of the results, we can perform an apost hoc test to check if one of the techniques can be singled out. For this pourpose we use the Nemenyi test. The Nemenyi statistic is obtained as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{4.5}$$

In our case with $k = 7$ methods to compare and $N = 17$ experiments the critical value for a 90% of confidence is $CD = 1.33 \cdot \sqrt{42}102 = 0.9855$. This result shows that there exists a statically significant difference between using two groups of methods. The group of the top performers include the Evolutionary Minimal, Forest-ECOC and One vs. One codings, versus the group of the worst performers which includes the Binary Minimal, One vs. All and DECOC codings. These results are very promising since following the Occam razor's principles in equal conditions simpler models tend to be more useful. In addition, analyzing the mean rank we find that Evolutionary Minimal approaches are the two first in rank followed by Forest-ECOC and One vs. One coding. This result is very satisfactory and encourages the use of the Minimal approach since similar (or even better) results can be obtained with far less number of classifiers, in comparison to standard approaches.

Moreover, the evolutionary versions of Minimal designs outperform the Binary Minimal approach in most of the experiments. This result is expected since the evolutionary version looks for a Minimal ECOC matrix configuration that minimizes the error over the training data, increasing the generalization capability of the system. In particular, the advantage of the evolutionary version over the binary one is more significant when the number of classes increases, since more Minimal matrices are available for optimization.

On the other hand, possible reasons why the evolutionary Minimal ECOC design obtains similar or better performance results than the rest of approaches with far less number of dichotomizers can be the few classifiers considered for tuning, and the use of all the classes in balanced binary problems, which can help the system to increase generalization if a good decision boundary can be found by the classifier. Note that the One vs. One classifier looks for binary problems that split just two classes. In those cases, though good and fast solutions could be found in training time, the use of less data does not assure a high generalization capability of the individual classifiers.

In terms of testing time, the reduction of dichotomies defined by Minimal approaches compared to predefined or problem-dependent strategies, shown in Table 4.1, has the effect of significantly reducing the testing time, independently of the base classifier used. Observe the great difference in terms of the number of classifiers between the Minimal approaches and the classical ones. The Minimal approaches obtain an average speed up improvement of 111% with respect to the One vs. All approach in testing time, meanwhile in the case of the One vs. One technique this improvement is of 489%.

Discussion

In order to provide a complete description of the method, in this section we analyze its computational training and testing cost.

Let us consider the average computational cost for optimizing a certain dichotomizer as a constant value. Analyzing the number of dichotomizers produced by each coding design, we find that for the One vs. One coding we have $\binom{k}{2}$ dichotomizers to optimize, and thus, the computational cost of One vs. One coding is $O\left(\binom{k}{2}\right)$. In the case of One vs. All coding, we have k dichotomizers to be optimized, and thus, its computational cost is $O(k)$. In the case of Evolutionary Minimal codings (GA & PBIL) we have $O(G \cdot I \cdot log_2(k))$, where I represents the number of individuals in the genetic optimization (which is constant along generations) and G represents the number of generations of the genetic algorithm. Undoubtedly, this cost may be greater than the cost of the other techniques. Nevertheless, with the introduction of a cache of optimized dichotomizers each dichotomizer is optimized only once and its parameters are stored for future usage. In this sense, since the number of partitions of the classes is finite, as the algorithm progresses, the number of dichotomizers to be optimized exponentially decreases. Thus, with this approximation procedure the value of G tends to one. And in an optimal case the computational complexity becomes $O(I \cdot log_2(k))$.

On the other hand, to analyze the computational cost at the test step, we provide the average number of Support Vectors (SV) generated for each coding design. Observe that the testing time is proportional to the number of SVs [111]. Figures 4.4 and 4.5 show the number of SVs per coding design and dataset.

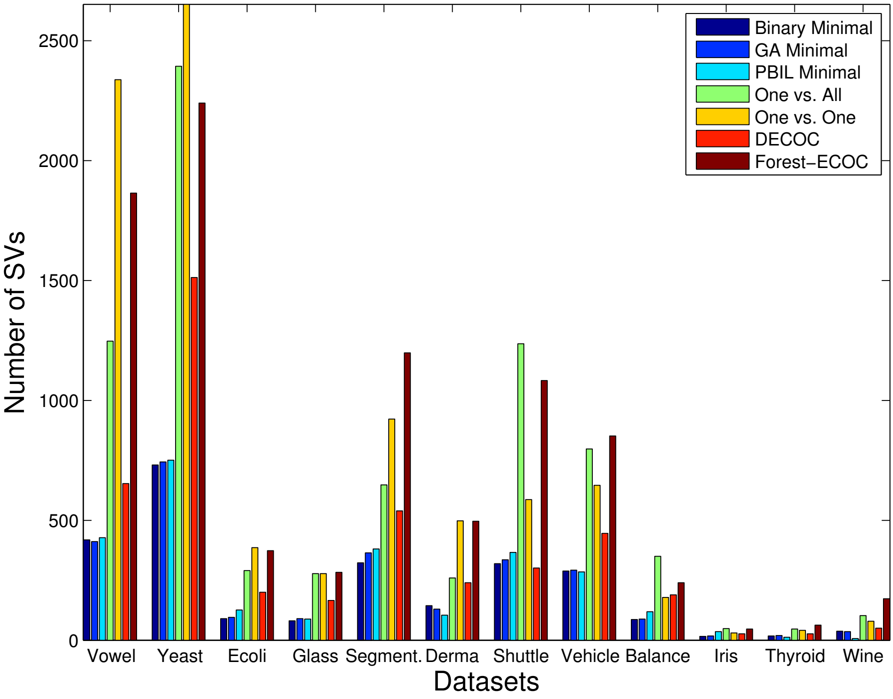


Figure 4.4: Number of SVs for each method on UCI datasets.

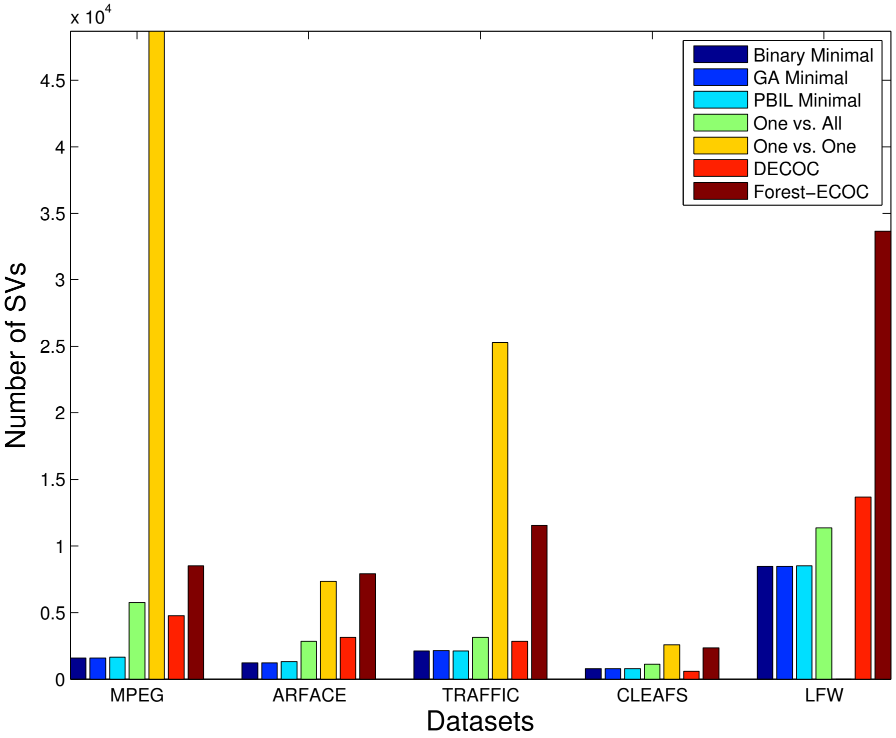


Figure 4.5: Number of SVs for each method on Computer Vision datasets.

Analyzing the empirical results shown in Figures 4.4 and 4.5, we find that the number of SVs defined by Minimal approaches is significantly smaller than the number defined by the rest of strategies. This difference becomes even more significant when comparing Minimal approaches with standard predefined coding designs, such as One vs. All and One vs. One, where in most cases the number of SVs defined by Minimal approaches is one order of magnitude smaller.

When comparing Minimal approaches (i.e. Binary, GA and PBIL) no statistical significance is found. This is an expected result since they define the same number of balanced dichotomizers, and thus, each base classifier is responsible of learning boundaries between bi-partitions of the complete set of categories. In this sense, we expect Minimal approaches to define a very similar number of SVs. Finally, Minimal approaches define a more reduced number of SVs than other standard predefined or even problem-dependent strategies. Concretely, evolutionary Minimal approaches define a compact but yet discriminate enough number of SVs, obtaining comparable or even better results than other coding designs while dramatically reducing the testing time.

4.3 Learning ECOCs using ECOC-compliant Genetic Algorithms

4.3.1 Genetic Algorithms in the ECOC Framework

As seen in previous sections, Genetic Algorithms are stochastic optimization processes based on Darwin's evolution theory. These processes start with a set of individuals which represent a set of random points in the search space. Each individual has a fitness value, which denotes the adaptation to the environment. Individuals are transformed through crossover and mutation operators along generations, improving their adaptation to the environment. Commonly, the crossover operator guides the optimization process to parts of a search space which optimize the fitness value. On the other hand, the mutation operator is responsible of not letting the algorithm converge to local minima. Recent literature on applying GAs to the ECOC framework is summarized in the following paragraphs.

- **Minimal Design of Error-Correcting Output Codes [10]:** In this work we proposed a standard GA to optimize an ECOC coding matrix $\mathbf{M} \in \{-1,+1\}^{k\times l}$, where $l = \lceil log_2 k \rceil$. In addition, the evaluation of each individual is obtained by means of its classification error over the validation set. In this work, the scattered crossover operator is used. Mutation is implemented using a Gaussian distortion over the selected gene. The achieved results are comparable with most of the state-of-the-art ECOC approaches with a reduced code length.
- **Evolving Output Codes for Multiclass Problems [62]:** This work proposed the use of the Cross generational elitist election, Heterogeneous recombination and Cataclysmic mutation (CHC) Genetic Algorithm [52] to optimize a Sparse Random ECOC matrix [4] $\mathbf{M} \in \{-1,0,+1\}^{k\times l}$ where $l \in [30,50]$. In this case, the code length is fixed in the interval $[30,50]$ independently of the number of classes, which seems to be a non well founded approach since it disagrees with a large body of literature [4, 10, 106, 111, 125]. It is interesting to note that the evaluation of individuals is the aggregation of different aspects of the ECOC coding matrix such as distance between rows and columns or dichotomizer performances.
- **Evolutionary Design of Multiclass Support Vector Machines [89]:** In this work the authors propose a Genetic Algorithm to optimize a Sparse Random [4] coding matrix $\mathbf{M} \in \{-1,0,+1\}^{k\times l}$, where $l \in [\log_2 k, k]$. The evaluation of each individual (ECOC coding matrix) is performed using the classification error over validation subsets. The crossover operator considered is the exchange of a set of $i \in [1,l']$ dichotomies between individuals,where $l' < l$. The considered mutation operator has four variants which depend on how values in the coding matrix are changed. An interesting point is that operator variants are chosen based on an historic of its performance in previous generations.

Despite these works, the problem of optimizing an ECOC coding matrix $\mathbf{M}$ presents some issues that must be carefully reviewed.

The first one is the uncontrolled generation of non-valid individuals (see Equations 2.7, 2.8, and 2.9). This issue has been treated in state-of-the-art works either by automatically setting the fitness value of non-valid individuals to be lower than the worst valid individual value, and thus, letting the algorithm converge to valid solutions, or by simply rejecting non-valid individuals both being valid options used in most evolutionary frameworks. Nevertheless, it is easy to see that when tackling the problem of minimizing the search space of ECOC matrices, the mentioned approximation is inappropriate, since the algorithm may never generate a valid individual. This is due to the particular structure and properties of ECOC matrices. The responsibles of generating of new individuals along the optimization process are crossover and mutation operators. Therefore, one may argue that operators used by state-of-the-art approaches are not suitable for the problem of optimizing an ECOC coding matrix $\mathbf{M}$, since they overlook the ECOC structure. Thus, they have to be redefined to comply with ECOC properties.

The second issue is how the optimization process is guided to parts of the search space that optimize the fitness of the individuals. In this sense, not only the constraints of individuals have to be taken into account when designing crossover and mutation operators but also how these individuals improve their adaptation through those operators, allowing the process to converge in fewer generations. On the other hand, designing operators that dramatically reduce the stochastic search may imply premature convergence to local minima.

Commonly, when making use of optimization processes in the ECOC framework the first step to perform is the estimation of the ECOC search space cardinality. Assume an k-class problem to be treated, then the ECOC framework will construct a matrix $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ in which k codewords will be chosen from the 3^l codes available. Following Newton's binomial this could be expressed as $\binom{3^l}{k}$. Nevertheless, taking into account the constraints defined in Equations 2.8 and 2.9 a matrix M and its opposite (swapping all $+1$ by -1 and vice-versa) are equivalent since they define the exact same partitions of the data. In this sense, the number of possible ECOC coding matrices is shown in Equation 4.6.

$$\#M = \frac{\binom{3^l}{k}}{2} = \frac{3^l!}{2 \cdot k! \ \cdot (3^l - k)!} \tag{4.6}$$

Furthermore, this search space is non-continuous since a change in a single position of the ECOC matrix $\mathbf{M}$ can break the ECOC matrix constraints. In addition to the huge cardinality of the non-continuous neither differentiable search space, [34] showed that the computation of an optimum ECOC matrix given the set of all possible dichotomizers $\mathbb{H}$ is *NP-Complete.*

In the next section we describe the method which is able to deal with all these issues by taking into account the properties of the ECOC framework within the definition of the GA.

4.3.2 ECOC-Compliant Genetic Algorithm

In this section the novel ECOC-Compliant Genetic Optimization is presented. In order to provide a complete description of the method in Figure 4.6 we show a scheme of the procedure.

Problem Encoding

Since our proposal redefines the standard crossover and mutation operators there is no need to be tied to standard encoding schemes. In this sense, our individuals are encoded as structures $I =< \mathbf{M}, \mathbf{C}, \mathbb{H}, \mathbf{p}, E, \delta >$, where the fields are defined as follows.

- The **coding matrix**, $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ where $l \geq \lceil \log_2 k \rceil$. Note that for the initial population $l = \lceil \log_2 k \rceil$, where l can grow along generations.
- The **confusion matrix**, $\mathbf{C} \in \mathbb{R}^{k \times k}$, over the validation subset. Let i and j be two classes of our problem, then the entry of $\mathbf{C}$ at the i-th row and the j-th column, defined as $c_{i,j}$, contains the number of examples of class i classified as examples of class j.
- The **set of dichotomizers**, $\{f^1, \dots, f^n\}$.

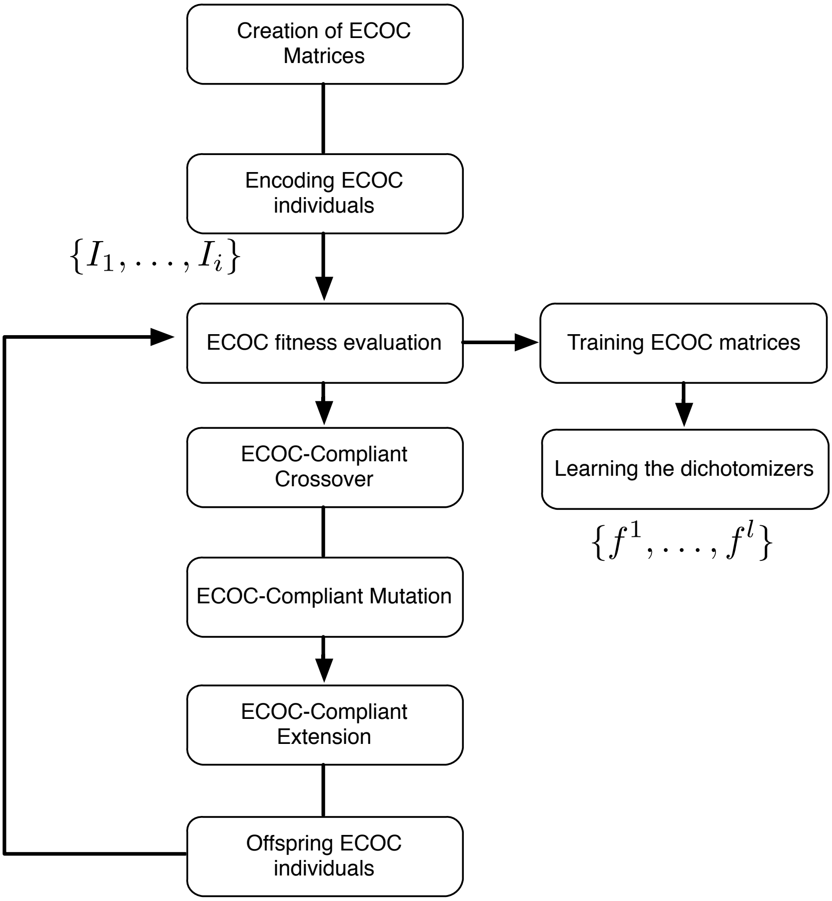


Figure 4.6: Diagram of the ECOC-Compliant Genetic Algortihm.

- The **performance of each dichotomizer**, $\mathbf{p} \in \mathbb{R}^l$. This vector contains the proportion of correctly classified examples over the validation subset for each dichotomizer. Note that this measure is not the performance of the overall multi-class problem but the one of the dichotomizer over the meta-classes defined by the correspondent dichotomy.
- The **error rate**, E, over the validation subset. This scalar is the proportion of incorrectly classified examples in the multi-class problem over the validation subset. Let the set of samples in the validation subset be $\mathbf{X}^{val} = \{\{\mathbf{x}_1, y_1\}, \ldots, \{\mathbf{x}_v, y_v\}\}$, then the calculus is shown in Equations 4.7 and 4.8.

$$E = \frac{\sum_{j=1}^{v} \mathcal{I}(\delta(\mathbf{M}, \mathbf{f}(\mathbf{x}_j), y_j)}{v} \tag{4.7}$$

$$\delta(\mathbf{M}, \mathbf{f}(\mathbf{x}_t)) = \operatorname*{argmin}_i \delta(\mathbf{m}^i, \mathbf{f}(\mathbf{x}_t)), \; i \in \{1, \ldots, k\}. \tag{4.8}$$

- The **decoding function**, δ. We use the Loss-Weighted decoding [47] of Equation 4.9, where $\mathbf{W} \in \mathbb{R}^{k \times l}$ is a matrix of weights and $\mathcal{L}$ is a loss function ($\mathcal{L}(\theta) = \exp^{-\theta}$ in our case).

$$\delta_{LW}(\mathbf{m}^i, \mathbf{f}(\mathbf{x})) = \sum_{j=1}^{l} w_{ij} \mathcal{L}(m_{ij} \cdot f^j(\mathbf{x}))) \tag{4.9}$$

Fitness Function

The fitness function measures the adaptation of each individual, and thus, is the one to be optimized. In this sense, the most common approach in state-of-the-art literature has been to evaluate an ECOC individual as the performance it obtains on the validation subset. Nevertheless, following Occam's razor principles, in the hypothetical situation in which two individuals obtain the same performance on the field the one showing a simpler model (which very often implies a smaller code length) tends to be the most suitable choice.

This general assumption can be redefined and used in the fitness function in order to penalize those individuals with a large code length. Let $\mathbb{I} = \{I_1, \ldots, I_i\}$ be a a set of individuals and I_k a ECOC individual encoded as shown in Section 4.3.2, then our fitness function is defined as shown in Equation 4.10.

$$\mathcal{F}(I_k) = E_{I_k} + \lambda l_{I_k} \tag{4.10}$$

This expression (similar to the one showed by regularized classifiers), serve us to control the learning capacity and avoid over-fitting.

There exists several ways of defining complexity in the ECOC framework. Nevertheless, the code length has been always in the core of this definition. Thus, we have adopted the term complexity as the number of dichotomies defined in the coding matrix $\mathbf{M}$, that is l.

ECOC-Compliant Crossover and Mutation Operators

In this section we introduce the novel ECOC-compliant crossover and mutation operators. These operators do not only take into account the restrictions of the ECOC framework (shown in Equations 2.7, 2.8 and 2.9) but are also designed in order to avoid a premature convergence to local minima without generating non-valid individuals, and thus, converging to satisfying results in fewer generations.

In our proposal, when performing the genetic optimization we have to take into account the effect of the operators used. In many Genetic Algorithms a trade-off between exploring a satisfying portion of the search space and converging quickly to a final population is desirable [70], thus avoiding the convergence to local minima. To achieve this goal in our proposal two versions of each operator were designed, the randomized and the biased. On one hand, the randomized version of each operator builds valid individuals with a random seed, which aims to accomplish the exploration of a satisfying portion of the search space. On the other hand, the biased version takes into account certain factors (i.e dichotomizer performances, confusion matrices, etc.) that imply the guidance of the optimization procedure to promising regions of the search space, where individuals may obtain a better fitness value.

- **ECOC-Compliant Crossover Algorithm**

In GAs one of the most important issues is how individuals are recombined in order to produce fitter offspring. In this sense, one would like to find an intelligent recombination method (known as crossover operator) that take profit of problem domain information in order to allow a faster convergence. Crossover operators are strongly defined for standard

GAs encodings schemes (such as binary encoding) and have been deeply studied in literature. Nevertheless, when facing problems in which individuals are constrained and standard encoding designs have to be redefined, we consider also the redefinition of the recombination procedure to be an unavoidable task. This consideration is given by the fact that using standard GAs in problems where individuals are constrained can lead to situations where the search space is enlarged due to the generation of non-valid individuals.

Consider a k-class problem and let I_F and I_M be two individuals encoded as shown in Section 4.3.2. Then the crossover algorithm will generate a new individual I_S which coding matrix $\mathbf{M}^{I_S} \in \{-1, 0, +1\}^{k \times l}$, $l = \min(\mathbf{M}^{I_F}, \mathbf{M}^{I_M})$ contains dichotomies of each parent. Therefore, the key aspect of this recombination is the selection of which dichotomies of each parent are more suitable to be combined. Taking into account the aim of avoiding the generation of non-valid individuals, we introduce a dichotomy selection algorithm that chooses l dichotomies that fulfil the constraints shown in Equations 2.7, 2.8, and 2.9.

The dichotomy selection algorithm generates a dichotomy selection order $\mathbf{t} \in \mathbb{N}^n$ for each parent, where $\mathbf{t}^I$ is the selection order of parent I and $\mathbf{t}^I_k$ is the value at the k-th position. However, this selection order can lead to a situation in which the l dichotomies chosen define an incongruence in the coding matrix, such as defining two classes with the same codeword. In such case, the dichotomy selection algorithm checks if the separation between codewords is congruent with the number of dichotomies left to add.

Theorem 4.3.1 describes the number of equivalent codes allowed to appear on a matrix that is being built to fulfil the ECOC properties in terms of rows (Equation 2.7). In this sense, when the final length of the ECOC matrix in terms of columns is known, we can determine the maximum number of equivalent codes allowed to appear when each extension dichotomy is appended to build the ECOC matrix.

Theorem 4.3.1 *Should* $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ *be a randomly distributed matrix. Then, the extension of* $\mathbf{M}^{k \times l}$ *to an ECOC coding matrix* $\mathbf{M} \in \{-1, 0, +1\}^{k \times (l+l')}$ *with k unequivocally defined rows, will be possible if and only if when including the i-th* ($0 \leq i \leq l'$) *extension dichotomy in* $\mathbf{M}$*,* $2^{(l'-i)}$ *repeated codewords of length* $t + i$ *are obtained at most.*

Proof Let a matrix $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ define R_l repeated codes (two codes $\mathbf{m}^i, \mathbf{m}^j$ are equivalent if $\delta_{AHD}(\mathbf{m}^i, \mathbf{m}^j) = 0$). Assume $\mathbf{M}^{k \times l}$ is to be extended to $\mathbf{M}^{k \times (l+l')}$ so that it fulfills Equation 2.7: $\min(\delta_{AHD}(\mathbf{m}^i, \mathbf{m}^j)) \geq 1$, $\forall i, j : i \neq j$, $i, j \in [1, \ldots, k]$.

Then, from Information theory $\lceil \log_2(R_l) \rceil$ is known to be the minimal number of extension bits needed to unequivocally split R_l codes. Therefore, if $\mathbf{M}^{k \times l}$ is extended with l' dichotomies, then $\lceil \log_2(R_l) \rceil \leq l'$ dichotomies are needed to assure that Equation 2.7 holds. When the first of the l' dichotomies is added, then $l' - 1$ dichotomies will be used to split the remaining set of repeated codes (R_{l+1}). As in the former case, $\lceil \log_2(R_{l+1}) \rceil \leq l' - 1$ are needed. Accordingly, when the second dichotomy is appended $\lceil \log_2(R_{l+2}) \rceil \leq l' - 2$. Generalizing, $\mathbf{M}^{k \times l}$ will only be extensible to a valid ECOC matrix $\mathbf{M}^{k \times l + l'}$ if when adding the i-th dichotomy $\lceil \log_2(R_{l+i}) \rceil \leq l' - i$. Thus, $2^{l'-i}$ repeated codewords are obtained at most when adding the i-th extension dichotomy.

Following Theorem 4.3.1 the i-th dichotomy will be only added if it splits the existing codewords to define $R_{l+i} \leq 2^{(l'-i)}$ different codes. However, in a certain iteration it may happen that there are no existing dichotomies in both parents that accomplish the split criteria. In this situation, a new dichotomy is generated in order to ensure the ECOC properties. We define the ECOC-compliant crossover algorithm as shown in Algorithm 1.

Two ECOC crossover algorithm variants are proposed, which have an equal probability of being executed. In the first one, which is the randomized version, the dichotomy selection

```
Data: $I_F$,$I_M$
Result: $I_S$
$l := \min(\mathbf{M}^{I_F}, \mathbf{M}^{I_M})$ // Minimum code length among parents
$\mathbf{t}^{I_F} \in \mathbb{N}^l = selorder(I_F)$ // Dichotomy selection order of $I_F$
$\mathbf{t}^{I_M} \in \mathbb{N}^l = selorder(I_M)$;
$cp := I_F$ // Current parent to be used
$\mathbf{M}^{I_S} := \emptyset$ // Coding matrix of the offspring
for $i \in \{1, \ldots, l\}$ do
    for $j \in \{1, \ldots, l_{cp}\} : \mathbf{t}_j^{cp} \neq \emptyset$ do
        $f := 0$ // Valid dichotomy search flag
        if $calcRepetitions$ $(\mathbf{M}^{I_S}, \mathbf{m}_{\mathbf{t}_j^{cp}}) \leq 2^{(l'-i)}$ then
            $\mathbf{m}_i := \mathbf{m}_{\mathbf{t}_j^{cp}}$ // Inheritance of dichotomies
            $f^i := f^{\mathbf{t}_j^{cp}}$ // Inheritance of dichotomizer
            $p_i := p_{\mathbf{t}_j^{cp}}$ // Inheritance of performance
            $\mathbf{t}_j^{cp} := \emptyset$ // Avoid using a dichotomy twice
            $f := 1$ // Valid dichotomy found
            $break$;
        end
    end
    if $!f$ then
        $\mathbf{m}_i := generateCol(\mathbf{M}^{I_S})$ // If non ECOC matrix can be built
        $f^i := \emptyset$;
        $p_i := \emptyset$;
    end
    if $cp = I_F$ then
        $cp := I_M$ // Dichotomy inheritance parent switch
    else
        $cp := I_F$;
    end
end
```

Algorithm 1: ECOC Crossover Algorithm.

order is randomly generated, and thus, it generates a random ECOC individual that ensures to fulfill Equations 2.7, 2.8, and 2.9. In the second one, the biased version, the dichotomy selection order is based on dichotomizer performance, and thus, dichotomizers that show a higher performance have more chances of being selected. These two variants of crossover provide us a trade-off between covering an enough portion of the search space and guiding the optimization process to a population with minimal values of the fitness function. An example of the ECOC-compliant crossover operator is shown in Figure 4.7.

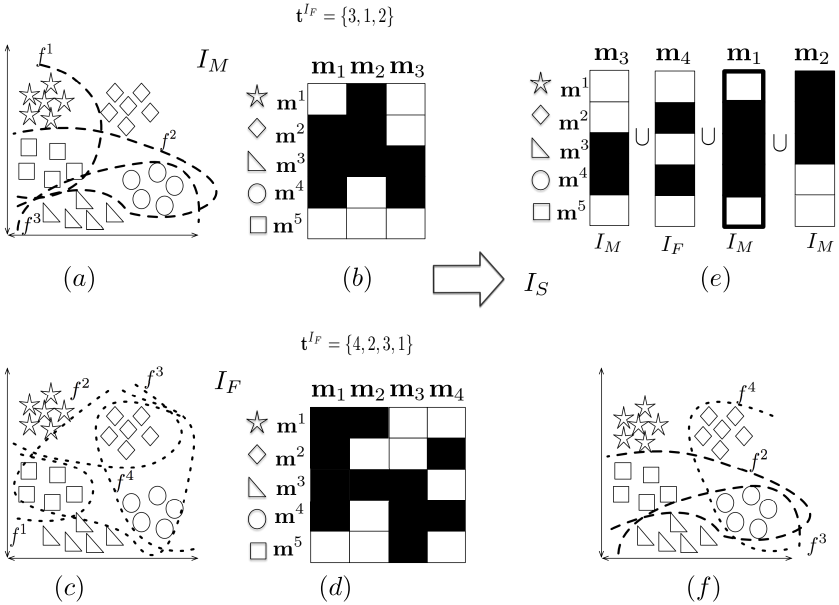


Figure 4.7: Crossover example for a 5-class toy problem. (a) Feature space and trained classifiers for parent I_M. (b) ECOC coding matrix of parent I_M. (c) Feature representation and boundaries for parent I_F. (d) Coding matrix of I_F. (e) ECOC coding matrix composition steps for the offspring I_S. (f) Feature space and inherited classifiers for I_S.

In the crossover example shown in Figure 4.7 two individuals I_M and I_F are combined to produce a new offspring I_S[2]. The crossover algorithm generates a dichotomy selection order $\mathbf{t}$ for each parent. The first parent from which a dichotomy is taken is I_M, and $\mathbf{m}_3$ is valid since $r \leq 2^{(3-1)} = 4$, and it only defines three codes without separation ($\mathbf{m}^1, \mathbf{m}^2$, and $\mathbf{m}^5$). Once this step is performed, the parent is changed, and the following dichotomy will be extracted from I_F based on its selection order τ^{I_F}. In this case, $\mathbf{m}_4$ is valid since $l \leq 2^{(3-2)} = 2$ and $\mathbf{m}_3$ of I_M together with $\mathbf{m}_4$ of I_F define only two equivalent codewords ($\mathbf{m}^1$ and $\mathbf{m}^5$). In the following iteration, the parent is changed again, and thus, I_M is used. Following $\mathbf{t}^{I_M}$ the dichotomy to use is $\mathbf{m}_1$, but if we apply Theorem 4.3.1 we find that $l \leq 2^{(3-3)} = 1$, and thus, $\mathbf{m}_1$ is unuseful. Since $\delta_{AHD}(\mathbf{m}^1, \mathbf{m}^5) = 0$, $\mathbf{m}_1$ can not be considered as an extension dichotomy, and therefore, the next dichotomy to use is $\mathbf{m}_2$, which satisfies Equation 2.7 defining a valid ECOC coding matrix.

- **ECOC-Compliant Mutation Algorithm**

[2] Note that for applying Theorem 4.3.1 in this example we consider $l' = 3$.

Historically, mutation operators have been responsible of not letting the algorithm converge to local minima. In literature, these operators have been defined for standard encoding designs (such as binary encoding). Nevertheless, when individuals are not encoded following standard schemes these operators have to be redefined in order to completely fulfill their purpose.

Let an individual I encoded as shown in Section 4.3.2 to be transformed by means of the mutation operator. This operator will select a set of positions $\mu = \{m_{ij}, \ldots, m_{kl}\}, i, k \in \{1, \ldots, j\}, j, l \in \{1, \ldots, l\}$ of $\mathbf{M}^I$ to be mutated. The value of these positions is changed constrained to the set $\{-1, +1, 0\}$. Two variants of this algorithm are implemented depending on how the positions in μ are chosen and how **M**is recoded. The first is defined as the randomized ECOC mutation algorithm shown in Algorithm 2. In this version the set of positions μ is randomly chosen. Once μ is defined, the positions are randomly recoded to one of the three possible values in $\{-1, +1, 0\}$. However, note that the mutation of values may lead to a situation in which the matrix **M** does not fulfil the ECOC constraints. To avoid this effect, we check the ECOC matrix at each bit mutation in order to ensure that a valid ECOC individual is generated, if a certain bit mutation generates a non-valid individual this particular bit mutation is disregarded.

```
Data: $I_T, mt_c$
// Individual and mutation control value
Result: $I_X$
$\mu = \{\{i,j\}, \ldots, \{k,l\}\},\ i,k \in \{1,\ldots,k\},\ j,l\ \in \{1,\ldots,l\}$ ;
$\mu =$ getRandomPositions($\mathbf{M}^{I_T}, mt_c$) // Select the position in $\mathbf{M}^{I_T}$ for mutation
;
for $\{i,j\} \in \mu$ do
    switch $m_{ij}$ do
        // If the value selected for mutation is 0 it might turn $+1$ or $-1$
        case $m_{ij} = 0$
            $r =$Random(0,1); if $r > 0.5$ then
                $m_{ij} := +1$;
            else
                $m_{ij} := -1$;
            end
        end
        case $m_{ij} = -1$
            $r =$Random(0,1)// Obtain a random value in $[0,1]$
            if $r > 0.5$ then
                // Equiprobablity of selecting the remaning values
                $m_{ij} := +1$;
            else
                $m_{ij} := 0$;
            end
        end
        case $m_{ij} = +1$
            $r =$Random(0,1);
            if $r > 0.5$ then
                $m_{ij} := 0$;
            else
                $m_{ij} := -1$;
            end
        end
    endsw
end
$\mathbf{M}^{I_X} = \mathbf{M}$;
```

Algorithm 2: ECOC-Compliant Randomized Mutation Algorithm.

In the latter, defined as the ECOC-Compliant biased mutation algorithm, the set of positions μ is chosen taking into account the confusion matrix **C**. In this sense, the mutation algorithm will iteratively look for the most confused categories in the confusion matrix

$\{i, j\} = \underset{i,j}{\text{argmax}}\ c_{ij} + c_{ji}$. Once these classes are obtained, the algorithm will transform the bits valued 0 of codewords $\mathbf{m}^i$ and $\mathbf{m}^j$ in order to increase the distance $\delta_{AHD}(\mathbf{m}^i, \mathbf{m}^j)$, and thus, increasing their correction capability, while keeping a valid ECOC matrix. The ECOC-Compliant biased mutation algorithm is shown in Algorithm 3.

```
Data: $I_T, mt_c$
// Individual and mutation control value
Result: $I_X$
$\mathbf{C}^{I_T}\mathbb{R}^{k \times k}$ // Confusion matrix of $I_T$
$ct := 0$// Number of recoded bits of $M^{I_T}$
while $ct < mt_c$ do
    $\{i, j\} := \underset{i,j}{\text{argmax}}\ c_{ij} + c_{ji}\ \forall i, j : i \neq j$;
    for $b \in \{1, \ldots, l\}$ do
        if $|m_{ib}| + |m_{jb}| \leq 1$ and $ct < mt_c$ then
            if $m_{ib} = 0$ and $m_{jb} = 0$ then
                $m_{ib} := +1$ // Invert both bits valued 0
                $m_{jb} := -1$;
            else
                if $m_{ib} = 0$ then
                    $m_{ib} := -m_{jb}$ // Invert bit valued 0
                else
                    $m_{jb} := -m_{ib}$;
                end
            end
            $ct := ct + 1$;
        end
    end
    $c_{ij}^{I_T} := 0$, $c_{ji}^{I_T} := 0$;
end
```

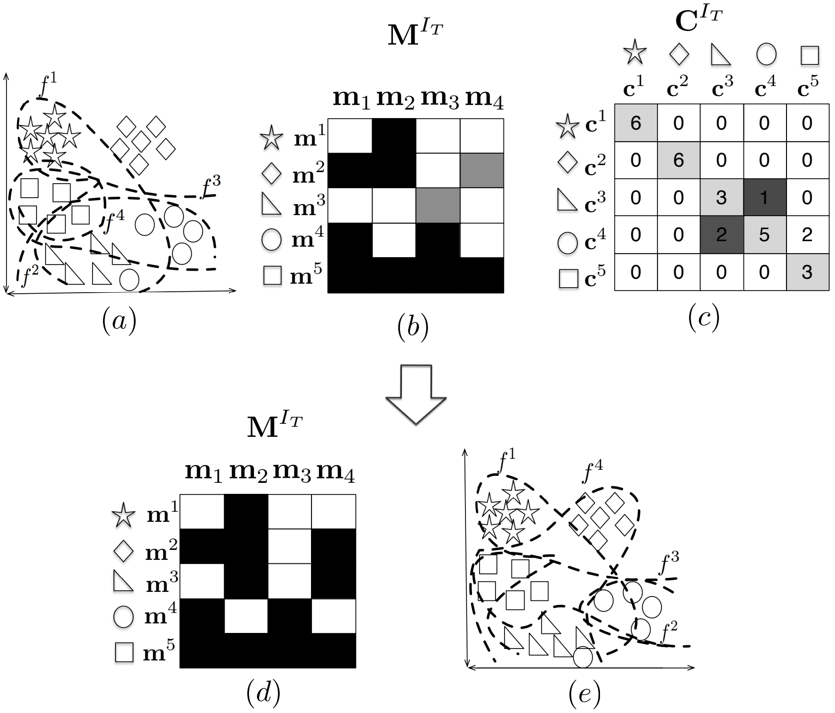
Algorithm 3: Biased ECOC-Compliant Mutation Algorithm.

In Figure 4.8 an example of the biased mutation algorithm is shown. Let I_T be an individual encoded as shown in Section 4.3.2. The confusion matrix $\mathbf{C}^{I_T}$ has its non-diagonal maximum at $c_{43} + c_{34}$. Then codewords $\mathbf{m}^4$ and $\mathbf{m}^3$ are going to be mutated. The 0 valued bits of this codewords are changed in order to increment $\delta_{AHD}(\mathbf{m}^4, \mathbf{m}^3)$, and thus, incrementing also the correction capability between them. At the following iteration c_{43} is not taken into consideration and the procedure will be repeated with $\mathbf{m}^5$ and $\mathbf{m}^4$ which are the following classes that show confusion in $\mathbf{C}$.

Problem-Dependent Extension of ECOCs

In related works that used GAs to optimize the ECOC matrices the length was fixed in a certain interval and the crossover operators where the ones responsible for obtaining reduced or large codes [62, 89]. Nevertheless, we consider that from the ECOC point of view the length of the code is a crucial factor that has to be addressed separately, since the length of the code matrix has a direct relationship to its correction capability. In this sense, as stated in Section 4.3.2 our initial population is based on the coding scheme proposed by [10], that is, the use of a Minimal ECOC coding matrix $\mathbf{M} \in \{-1, 0, +1\}^{k \times \lceil \log_2(k) \rceil}$. Nevertheless, when analyzing the Minimal ECOC matrix a lost of correction capability is found. Let $\mathbf{M}$ be a ECOC coding matrix, then:

$$\rho = \min\left(\frac{\delta(\mathbf{m}^i, \mathbf{m}^j) - 1}{2}\right),\ \forall i, j \in \{1, \ldots, k\},\ i \neq j. \tag{4.11}$$



	$\mathbf{c}^1$	$\mathbf{c}^2$	$\mathbf{c}^3$	$\mathbf{c}^4$	$\mathbf{c}^5$
$\mathbf{c}^1$	6	0	0	0	0
$\mathbf{c}^2$	0	6	0	0	0
$\mathbf{c}^3$	0	0	3	1	0
$\mathbf{c}^4$	0	0	2	5	2
$\mathbf{c}^5$	0	0	0	0	3

Figure 4.8: Mutation example for a 5-class toy problem. (a) Feature space and trained dichotomizers for and individual I_T. (b) ECOC coding matrix of I_T. (c) Confusion matrix of I_T. (d) Mutated coding matrix. (e) Mutated feature space with trained dichotomizers.

Therefore, we obtain a null correction capability $\rho = 0$ for the Minimal ECOC design, since for this ECOC matrices:

$$\min(\delta_{AHD}(\mathbf{m}^i, \mathbf{m}^j)_{\forall i,j:i \neq j}) = 1,\ i,j \in [1,\ldots,k]. \tag{4.12}$$

This means that in Minimal ECOC coding schemes, a sample $\mathbf{x}$ will be misclassified if just a dichotomizer f^i misses its prediction. Although this coding design has proved to be fairly effective when its properly tuned, we believe that an extension of such is needed to properly benefit from Error-Correcting principles. However, this extension is not only motivated by the null correction capability issue. The confusion between categories is also a determinant factor when extending ECOC designs [48], since one would like to focus dichotomies in those categories which are more difficult to be learnt.

We propose a novel methodology to extend ECOC designs based on the confusion matrix, aiming to focus the extension dichotomies in those categories which are more difficult to be learnt, and thus, show a greater confusion. This methodology defines to types of extensions, the One vs. One extension and the Sparse extension, which have the same probability of being executed along the optimization process. In the former, the ECOC coding matrix $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ will be extended with a dichotomy $\mathbf{m}_{l+1}$ which will have 0 values except for those two positions $\mathbf{m}_i$ and $\mathbf{m}^j$ which correspond to the categories $\{i, j\}$ that maximize the confusion $\{i, j\} = \underset{i,j}{\text{argmax}}\ c_{ij} + c_{ji}$. In the second, the Sparsity Controlled

Extension shown in Algorithm 4 follows the scheme in which two categories $\{i, j\}$ that maximize the confusion are discriminated. Nevertheless, as high effort to obtain both reduced and powerful codes is performed, one may want to extend $\mathbf{M}$ controlling the sparsity of $\mathbf{m}_{l+1}$. Hence, generating a dichotomy that is focused on certain categories but also increments the correction capabilities of $\mathbf{M}$.

Picture the case in which a dichotomy $\mathbf{m}_{l+1,i} = 0,\ \forall i \in \{1, \ldots, k\} \setminus \{\mathbf{m}_{l+1,r}, \mathbf{m}_{l+1,s}\}$. The Sparse extension algorithm will set $\mathbf{m}_{l+1,i}$ to $\{-1, +1\}$, based the confusion of class i with r and s. In this case, a $\mathbf{m}_{l+1,i}$ will be valued $v \in \{-1, +1\}$ if $\mathbf{m}_{\underset{iv}{\text{argmin}}\ c_{ir}, c_{is}} : v \in \{r, s\}$ is valued v. An example of Sparsity extension procedure is shown in Figure 4.9.

```
Data: $I_T, sp_c$
// Individual and sparsity control value
Result: $I_X$
$\mathbf{C}^{I_T}_{k \times k}$ // Confusion matrix of $I_T$
$\{i, j\} := \underset{i,j}{\text{argmax}}\ c_{ij} + c_{ji}\ \forall i, j : i \neq j$;
$ct := 0$ // Recoded bit counter of $\mathbf{M}^{I_T}$
$\mathbf{m}_{l+1,i} = \omega$ // Where $\omega \in \{+1, -1\}$
$\mathbf{m}_{l+1,j} = -\omega$;
for $b \in \{1, \ldots, k_{I_T}\} \setminus \{i, j\} : \underset{b}{\text{argmin}}\ c_{bi} + c_{ib} + c_{bj} + c_{jb}$ and $ct < sp_c$ do
    if $c_{bi} > c_{bj}$ and $m_{k+1,j} = \omega$ then
        // Give an inverse value to the bit of the class which is most confused with $i$ or $j$
        $\mathbf{m}_{k+1,b} = \omega$;
    else
        $\mathbf{m}_{k+1,b} = -\omega$;
    end
    $ct = ct + 1$;
end
```

Algorithm 4: Sparsity Controlled Extension Algorithm.

Implementation Details

[111] stated that if dichotomizers are high capacity classifiers and are properly tuned, the code length can be reduced to obtain simpler models. Following this idea, we adopted Support Vector Machines with a Gaussian RBF Kernel (SVM-RBF) as our dichotomizer, since it proved to be a very powerful classifier in literature. Typically, training a SVM implies the selection of certain data points (Support Vectors) to build the boundaries. In the specific case of the SVM-RBF two parameters have to be tuned in order to reach for good performances. This parameters are the regularization parameter C and the kernel parameter γ, which are closely related to the data distribution. In literature, the main approach to choose these parameters is the use of cross-validation processes to find the best $\{C, \gamma\}$ pair over a discretization of the parameter space. However, some works have shown that GA's can be applied to this problem, since it can be seen as an optimization process [10, 90]. In this sense, we use a GA to determine the value of the $\{C, \gamma\}$ pair for every dichotomizer in H.

4.3.3 Experiments

In order to present the experimental results, we first introduce the data, methods, and evaluation measurements of the experiments.

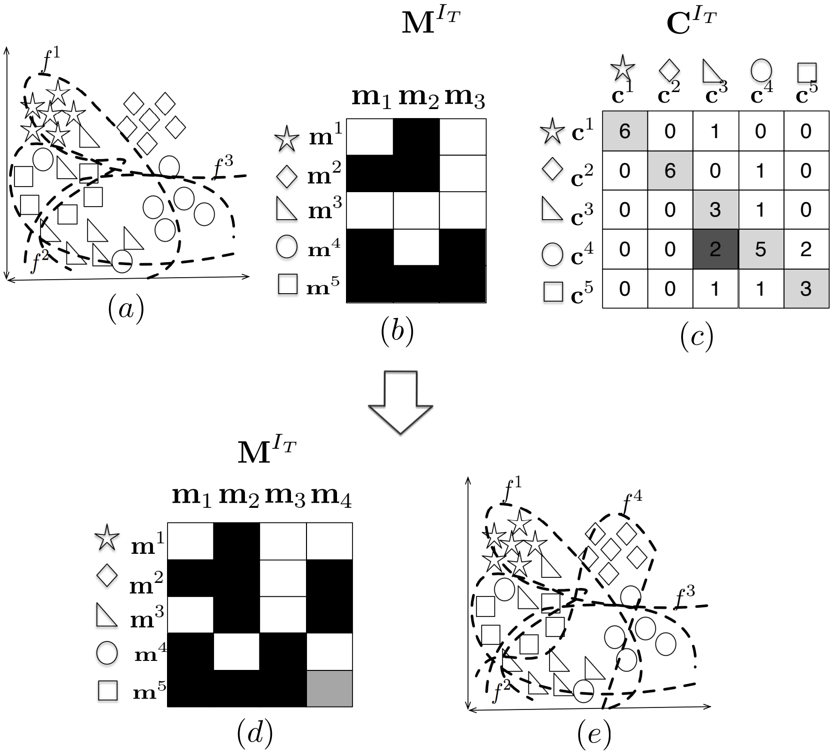


Figure 4.9: Sparsity extension example for a 5-class toy problem. (a) Feature space and trained dichotomizers for I_T. (b) ECOC coding matrix of I_T. (c) Confusion matrix of I_T. (d) Extended coding matrix. (e) Extended feature space with trained dichotomizers.

Data

The data used on these experiments are the following UCI datasets [8]: *Dermathology*, *Ecoli*, *Vehicle*,, *Segmentation*, *Glass*, *Vowel*, *Shuttle*, *Satimage*, and *Yeast*.

In addition we performed experiments in four challenging Computer Vision datasets: *Traffic Sign*, *ARface*, *Cleafs and Accidental*, and *MPEG7*. Detailed descriptions and characteristics for each dataset can be found in Appendix A.

Methods

We compare the One vs. One [125] and One vs. All [111] ECOC, DECOC [106] and Forest-ECOC [49] approaches with the novel ECOC-compliant genetic approach. Moreover, the approaches of Lorena et. al [89] and Pedrajas et. al [62] have been replicated in order to obtain a fair comparison with state-of-the-art ECOC GA methods. The Loss-Weighted decoding is applied at the decoding step [47]. The ECOC base classifier is the libsvm implementation of SVM with Radial Basis Function kernel [29].

Experimental Settings

For all experiments the base classifier used is an SVM with an RBF kernel. The optimization of its parameters is performed with a GA using a population of 60 individuals, using the operators defined by [10]. In addition for all evolutionary methods ([62], [89] and our proposal), the number of ECOC individuals in the initial population is set to $5k$, where k is the number of classes of the problem. The elite individuals is set to 10% of the population size. The stopping criteria is a stall activity of performance results during five generations.

Evaluation Measurements

The classification performance is obtained by means of a stratified five-fold cross-validation, and tested for the confidence interval with a two-tailed t-test. We also apply the Friedman and Nemenyi tests [38] in order to look for statistical significance among the obtained performances.

Experimental Classification Results

The classification results obtained for all the datasets considering the different ECOC configurations are shown in Table 4.2. The main trend of experimental results is that the One vs. One coding is the most successful coding in terms of performance, obtaining very good results in most of the datasets. Nevertheless, in certain situations coding designs with far less number of dichotomizers can achieve similar or even better results (i.e Ecoli, Yeast, and CLEAFS results). Moreover, taking into account the number of classifiers yielded per each coding design we can see how those codings that were optimized with a GA are far more efficient than the predefined ones. In addition, in order to compare the performances provided for each strategy, the table also shows the mean rank of each ECOC design considering the 26 different experiments (13 classification accuracies and 13 coding lengths). The rankings are obtained estimating each particular ranking r_{ij} for each problem i and each ECOC configuration j, and computing the mean ranking for each design as $\overline{r_j} = \frac{1}{N}\sum_i r_i^j$, where N is the total number of datasets. We also show the mean number of classifiers (#) required for each strategy. Furthermore, Table 4.3 shows the mean performance ranking and the mean performance per classifier ranking, which is computed as the rank of $PC = \frac{\sum_{i=1}^{\mathcal{N}} 1-E^i}{\sum_{i=1}^{\mathcal{N}} n^i}$, where $1 - E_i$ is the performance obtained in the i-th problem and n^i is the length of the code in the i-th problem.

In order to reject the null hypothesis that the measured performance ranks differ from the mean performance rank, and that the performance ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows (following the notation of the seminal paper of Demsat et. al [38]):

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]. \tag{4.13}$$

In our case, with $k = 8$ ECOC designs to compare, $X_F^2 = 11.26$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2}. \tag{4.14}$$

Applying this correction we obtain $F_F = 1.1$. With eight methods and thirteen experiments, F_F is distributed according to the F distribution with 7 and 175 degrees of freedom.

The critical value of $F(7, 175)$ for 0.05 is 0.31. As the value of F_F is higher than 0.31 we can reject the null hypothesis.

Furthermore, we perform a Nemenyi test in order to check if any of these methods can be singled out [38], the Nemenyi statistic is obtained as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \tag{4.15}$$

In our case, for $k = 8$ ECOC designs to compare and $N = 26$ experiments the critical value for a 90% of confidence is $CD = 2.780 \cdot \sqrt{\frac{56}{156}} = 1.8$. In this case, since our approach is the best in rank but it intersects with Minimal ECOC, Pedrajas et. al, DECOC, and, Lorena et al. approaches, we can state that there is no statistically significant difference among these five approaches. However, since our approach uses less dichotomizers than any of the other approaches (except for the Minimal ECOC which obtains the lowest classification accuracy ranking) it can be considered as the most suitable choice.

Moreover, we have to take into account that although Binary Minimal ECOC, Pedrajas et. al, DECOC, and, Lorena et al. approaches intersect with our proposal, the number of SVs defined by these approaches are generally bigger than the number of SVs defined by our proposal. Therefore, their testing complexity is larger than the one showed by our method (see Section 4.3.4 for further details). As for the Minimal ECOC method, although the mean rank is close to our method we can see in Table 4.2 that the randomness of its optimization leads to a much lower classification performance.

These results support our thesis of using far less dichotomizers than standard techniques while searching for ECOC matrices with a extremely high efficiency. In addition, the general trend of the experiments shows that the proposed ECOC-Compliant GA improves the classification accuracy of methods at the same complexity level and reduces the computational complexity of methods with similar accuracy.

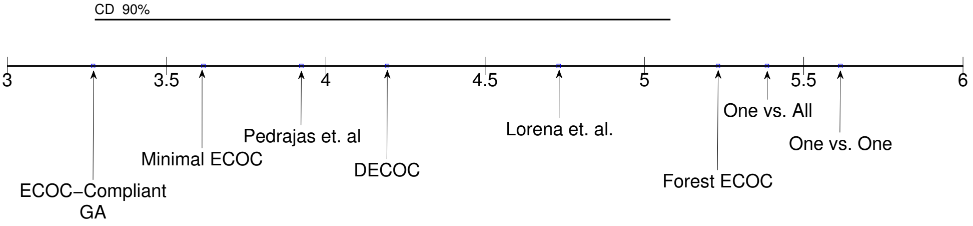


Figure 4.10: Critical difference for the Nemenyi test and the performance per classifier ranking values.

Table 4.2: Classification results and number of classifiers per coding design.

	ECOC-Compliant GA		Minimal ECOC		Lorena et al.	
Dataset	Perf.	#Class.	Perf.	#Class.	Perf.	#Class.
Vowel	64.7±13.4	5.6	57.7±22.2	4.0	69.3±11.3	9.2
Yeast	55.6±12.2	5.0	50.2±17.3	4.0	46.9±15.3	6.4
E.coli	**84.5±10.2**	3.0	80.5±9.7	3.0	83.1±13.2	5.2
Glass	50.1±22.8	5.0	38.4±23.4	3.0	45.2±21.9	6.4
Segment	96.8±1.2	5.0	66.9±3.4	3.0	97.1±1.4	5.6
Dermatology	96.3±3.1	3.8	96.0±4.2	3.0	96.5±2.9	3.8
Shuttle	74.8±13.2	4.0	72.5±25.3	3.0	73.6±13.2	4.6
Vehicle	81.1±10.3	3.0	72.5±13.3	2.0	82.0±12.2	5.6
Satimage	84.3±3.1	4.0	79.2±4.2	3.0	54.7±6.5	6.6
MPEG	84.4±2.8	7.0	89.3±3.9	7.0	84.4±1.3	7.0
ARFACE	76.5±4.8	5.4	76.0±5.7	5.0	84.2±3.2	8.4
TRAFFIC	84.1±3.6	6.0	90.8±2.6	6.0	92.3±2.9	6.8
CLEAFS	96.3±6.9	3.0	81.2±8.7	3.0	96.3±7.8	7.0
Rank & #Class.	4.5	4.5	6.2	**3.7**	4.4	6.3

	Pedrajas et al.		One vs. All		One vs. One	
Dataset	Perf.	#Class.	Perf.	#Class.	Perf.	#Class.
Vowel	55.7±18.3	7.0	**80.7±11.0**	11.0	78.9±13.2	28.0
Yeast	53.5±18.2	5.0	51.1±16.7	10.0	52.4±12.3	45.0
E.coli	83.1±13.3	3.0	79.5±10.3	8.0	79.2±12.3	28.0
Glass	56.1±25.7	5.0	53.9±23.5	7.0	**60.5±21.3**	15.0
Segment	96.8±1.7	3.0	96.1±2.2	7.0	**97.2±1.7**	21.0
Dermatology	95.7±2.3	4.0	95.1±1.3	6.0	94.7±2.3	15.0
Shuttle	68.5±17.2	4.0	**90.6±13.2**	7.0	86.3±14.2	21.0
Vehicle	79.6±15.7	3.8	74.2±11.2	3.0	83.6±9.6	6.0
Satimage	83.5±5.2	3.0	83.9±5.6	6.0	**85.2±7.9**	15.0
MPEG	84.7±2.6	7.0	87.8±3.4	70.0	**92.8±2.3**	2415
ARFACE	77.7±6.7	5.8	84.0±3.9	20.0	**96.0±2.8**	190.0
TRAFFIC	93.8±3.2	6.0	91.8±3.4	36.0	90.6±4.2	630.0
CLEAFS	94.9±6.3	3.0	80.8±4.8	7.0	84.2±6.7	21.0
Rank & #Class.	4.8	4.6	4.7	15.2	**3.2**	265.3

	DECOC		Forest ECOC	
Dataset	Perf.	#Class.	Perf.	#Class.
Vowel	66.8±12.8	8.2	70.2±10.3	30.0
Yeast	55.8±15.4	5.4	**56.1±13.2**	27.0
E.coli	69.5±9.7	4.2	75.2±8.9	21.0
Glass	55.0±21.3	5.4	46.9±21.8	15.0
Segment	97.0±2.3	4.6	97.1±1.3	18.0
Dermatology	**97.1±4.3**	2.8	96.0±2.1	15.0
Shuttle	77.1±13.2	3.6	84.4±12.1	18.0
Vehicle	**84.1±10.3**	4.6	81.7±13.3	9.0
Satimage	52.3±5.3	5.0	51.9±4.7	21.0
MPEG	83.4±2.5	69.0	88.9±4.3	207.0
ARFACE	82.7±4.3	19.0	85.6±4.2	147.0
TRAFFIC	86.2±2.9	35.0	**96.7±2.5**	105.0
CLEAFS	96.9±5.3	6.0	**97.1±4.2**	18.0
Rank & #Class.	4.3	14.2	3.4	50.0

4.3.4 Discussion

Regularization Analysis

In section 4.3.2 we defined the Fitness function of an ECOC individual as follows:

$$F(I_k) = E_{I_k} + \lambda l_{I_k}, \tag{4.16}$$

where λ is a user defined value that plays a regularization role for the ECOC matrix, similar to the control of learning capacity in regularized classifiers. In our proposal the value

Table 4.3: Mean rank per coding design.

	ECOC-Compliant GA	Minimal ECOC	Lorena et al.	Pedrajas et al.	One vs.All ECOC	One vs. One ECOC	DECOC	Forest ECOC
Perf. rank	4.5	6.2	4.4	4.8	4.7	**3.2**	4.3	3.4
#Class. rank	2	**1**	5	3	6	8	4	7
PC rank	**3.2**	3.6	4.7	3.9	5.3	5.6	4.1	5.2

of λ is estimated by a cross-validation procedure. In Figure 4.11 a cross-validation procedure to determine the λ value is shown for the *Vowel* dataset. In this procedure, the values of λ follow a logarithmic progression (from 0.01 to 1). It can be seen how the number of classifiers yielded by the proposal diminishes when λ increases (particularly between 0.01 and 0.05). Finding its minimum at $\lambda = 0.06$ which corresponds to the Minimal ECOC length [10].

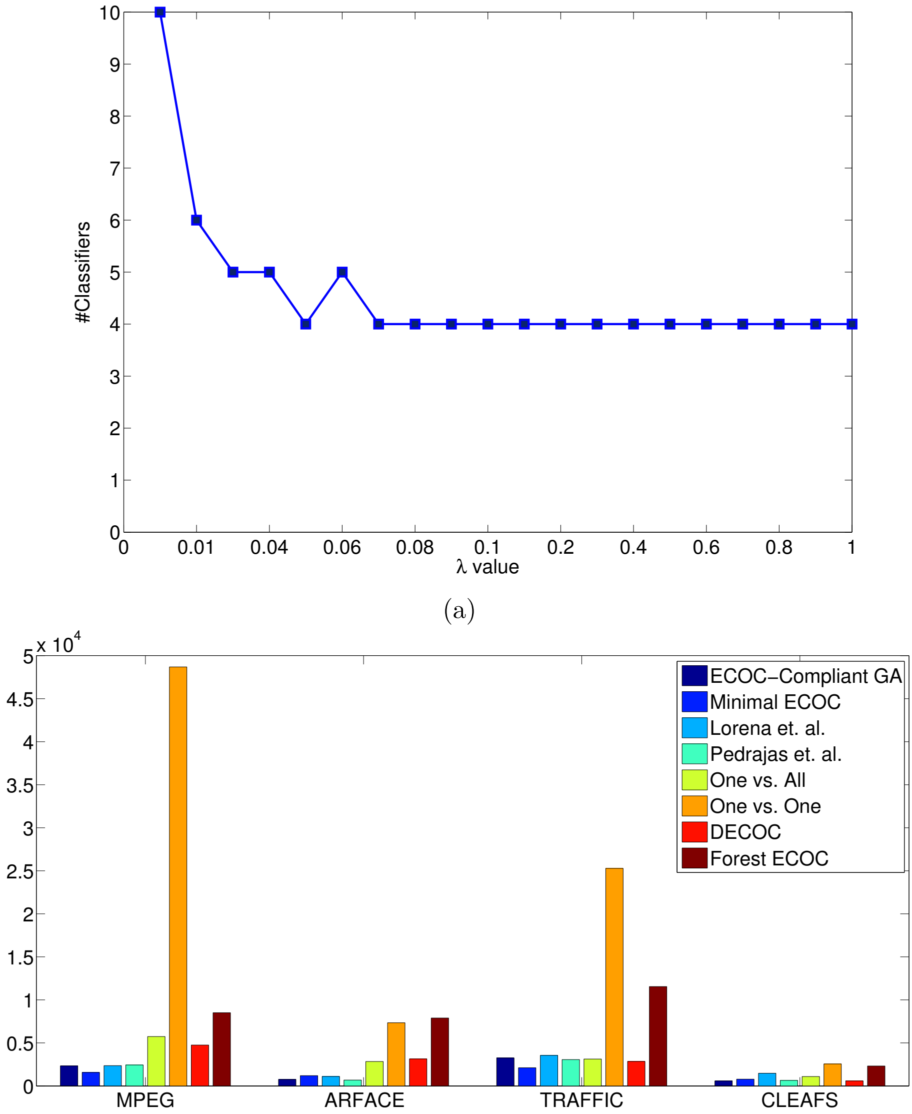
In this sense, in our experimental settings the lambda value was set to be in the middle point of the interval $[\lambda_{min}, \lambda_{max}]$, where λ_{min} is the smallest value of λ that yielded the smallest ECOC code length, and respectively for λ_{max}. Therefore, by performing this cross-validation setting of λ our proposal is able to find accurate models without a extremely high complexity in terms of the number of classifiers.

Complexity Analysis

In order to provide a complete description of the method, in this section we analyze its computational training and testing cost.

Let us consider the average computational cost for optimizing a dense dichotomizer (a dichotomizer in which all classes are taken into account) as a constant value. Then we find that in the case of One vs. All coding, we have k dichotomizers to be optimized, and thus, its computational cost is $O(k)$. In the case of evolutionary codings (Pedrajas et al., Lorena et al. and our proposal) we have $O(G \cdot I \cdot log_2(k))$, where I represents the number of individuals in the genetic optimization (which is constant along generations) and G represents the number of generations of the Genetic Algorithm. Undoubtedly, this cost may be greater than the cost of the other non evolutionary techniques. Nevertheless, with the introduction of a cache of optimized dichotomizers each dichotomizer is optimized once and its parameters are stored for future usage. In this sense, since the number of partitions of the classes is finite, as the algorithm progresses, the number of dichotomizers to be optimized exponentially decreases. Thus, with this approximation procedure the value of G tends to one. In consequence, in an optimal case the computational complexity becomes $O(I \cdot log_2(k))$.

In addition to the usual performance measures we also provide the number of Support Vectors (SVs) per coding scheme. Since this number is proportional to the complexity in the test step we can perform an analysis of what strategies show less complexity while still obtaining high performance. Figure 4.11(b) shows the number of SVs per coding design for the UCI data and Figure 4.11(b) shows the number of SVs for the Computer Vision problems. We can see that, in most cases, the four first columns of each dataset (corresponding to the Binary Minimal and all the evolutionary strategies) yield a lower number of SVs per model. Analyzing the empirical results shown in Figures 4.12 and 4.11(b), we find that the number of SVs defined by all three evolutionary strategies and the Binary Minimal approach are significantly smaller than the number of SVs defined by other strategies. This is due to the fact that all initial populations of evolutionary methods were set to follow a Minimal design [10], and thus, they are all expected to yield a similar number of SVs. However,



(a)

(b)

Figure 4.11: (a) Number of classifiers per λ value on the UCI *Vowel* dataset. (b) Number of SVs per coding design in the Computer Vision problems.

the new proposal defines a more reduced number of SVs than other standard predefined or even problem-dependent strategies. Defining a compact but yet discriminate enough number of SVs, obtaining comparable or even better results than other coding designs while dramatically reducing the testing time.

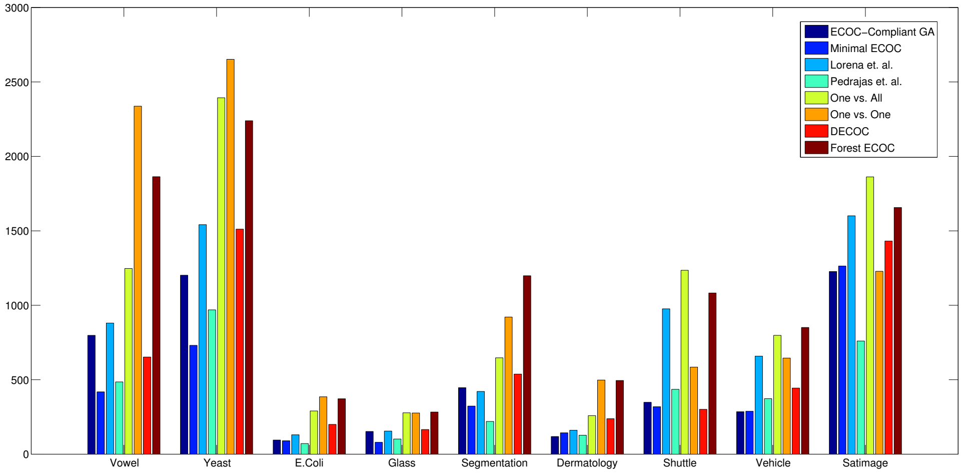


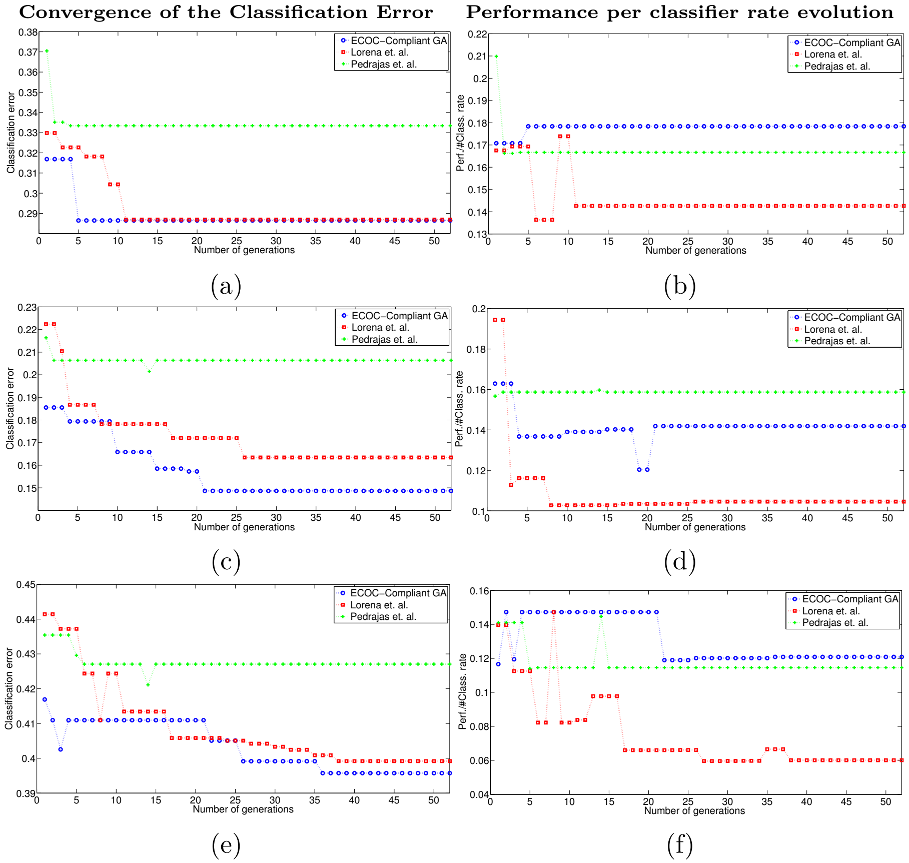
Figure 4.12: Number of SVs per coding design in the UCI datasets.

Convergence Analysis

This section is devoted to perform an analysis of convergence for the methods that use a GA to optimize the ECOC matrix ([62, 89] and our proposal). To properly perform this analysis we ran experiments for three UCI datasets (*Glass*, *Vowel* and *Yeast*), fixing the initial population to avoid the random initialization point issue (although equivalent ECOC matrices may yield different results due to the Genetic tuning of the SVM parameters). The number of generations was set to 50 and the rest of the experimental settings were the same as the ones described in Section 4.3.3.

In Figure 5.2 we show the evolution of the classification error, as well as the evolution of the performance per classifier rate for three UCI datasets. In Figures 5.2(a), 5.2(c) and 5.2(e) we can see how most of the times our proposal converges faster to better results than the state-of-the-art GA approaches. This fact is motivated by the redefinition of the operators that allows a fast exploration of the search space, without generating non-valid individuals. In addition, Figures 5.2(b), 5.2(d) and 5.2(f) show the evolution of the performance per classifier rate along generations. Figures 5.2(b) and 5.2(f) clearly show that our proposal yields models that are more efficient since we get a higher performance per classifier rate. Nevertheless, in Figure 5.2(d) the proposal of Pedrajas et al. obtains a higher rate. This is motivated by the fact that in the calculus of such rate both the classification error and the produced code length have the same weight. However, Figure 5.2(c) clearly shows that our method obtains better classification results.

Experimental results show that our proposal is able to converge faster to better results. In addition, the models yielded by the novel proposal are more efficient than the ones obtained by similar approaches. These results are obtained because of the redefinition of the crossover and mutation operators taking into account the theoretical properties of the ECOC framework. In this sense, our operators have a higher probability of finding good ECOC matrices than others since our search space is more reduced and the operators can guide the optimization procedure to promising regions of the search space.



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

Figure 4.13: (a) Classification error evolution for the *Glass* dataset. (b) Evolution of the Performance per classifier rate for the *Glass* dataset. (c)Classification error evolution for the *Vowel* dataset.(d) Evolution of the Performance per classifier rate for the *Vowel* dataset. (e)Classification error evolution for the *Yeast* dataset. (f) Evolution of the Performance per classifier rate for the *Yeast* dataset.

4.4 Conclusions

In this chapter we presented our contributions to learn an ECOC matrix using different types of Evolutionary approaches. In particular, for the Evolutionary Minimal ECOC method we presented a general methodology for the classification of several object categories which only requires $\lceil \log_2 k \rceil$ classifiers for a k-class problem. The methodology is defined in the Error-Correcting Output Codes framework, designing a Minimal coding matrix in terms of dichotomizers which univocally distinguish k codes. Moreover, in order to speed up the design of the coding matrix and the tuning of the classifiers, evolutionary computation is also applied. The results over several public UCI datasets and five multi-class Computer Vision problems with several object categories show that the proposed methodology obtains comparable (even better) results than state-of-the-art ECOC methodologies with far less number of dichotomizers. For example, the Minimal approach trained 10 classifiers to split

184 face categories, meanwhile the one-versus-all and one-versus-one approaches required 184 and 16836 dichotomizers, respectively.

Furthermore, we observed that operators for standard GA encoding fail to take into account the properties of ECOC matrices and their structure. Thus, we proposed to redefine the crossover and mutation operators to avoid the massive generation of non-valid individuals. Specifically, we presented a novel ECOC-Compliant Genetic Algorithm for the coding step in the ECOC framework. The proposed methodology redefines the usual crossover and mutation operators taking into account the properties of ECOC matrices. As a result, the search space is cropped, which causes the optimization to converge faster. Furthermore, a new operator which is able to increment the code length in a smart way was also introduced. The initial ECOC population followed a Minimal coding scheme, in which only $\lceil log_2(k) \rceil$ classifiers are needed to discriminate k classes, and as consequence, the methodology is able to find very efficient codes.

The ECOC-Compliant GA was tested on a wide set of datasets of the UCI Machine Learning Repository and over four challenging Computer Vision problems. For comparison purposes state-of-the-art ECOC GA schemes were replicated, as well as standard ECOC coding techniques. All the experiments were carried out using Support Vector Machines with an RBF kernel. Experimental results showed that the new proposal obtains significant improvements in comparison to state-of-the-art techniques. In particular, we analyzed the performance in terms of code length, and training and testing complexity. Those analysis showed that our proposal is able to find ECOC matrices with a high efficiency based on a trade-off optimization between performance and complexity obtained along the GA ECOC-Compliant optimization process.

Chapter 5

Learning ECOCs via the Error-Correcting Factorization

Problem-dependent ECOC designs in literature have always focused on the column picture of the ECOC matrix, that is, on the design of the binary problems which are coupled in the ECOC coding design. In order to look for the more suitable binary problems, several algorithms have been applied with successful results [49, 60, 62, 106, 141] and in previous chapters of this thesis we presented our contribution using GAs. However, we consider that analyizing the row picture (i.e. the distance between codewords) can shed more interesting results to analyze the Error-Correcting properties of an ECOC design, and yet surprisingly has not been exploited yet.

In this chapter, we present the Error-Correcting Factorization (ECF) method for factorizing a design matrix $\mathbf{D} \in \mathbb{R}^{k \times k}$ of desired 'error-correcting properties' between classes into a discrete ECOC matrix. The proposed ECF method is a general framework for the ECOC coding step since the design matrix is a flexible tool for error-correction analysis. In this sense, the problem of designing the ECOC matrix is reduced to defining the design matrix, where higher level reasoning may be used. For example, following recent state-of-the-art works one could build a design matrix following a *"hard classes are left behind"* spirit, boosting the boundaries of easily separable classes and disregarding the classes that are not easily separable. An alternative for building the design matrix is the *"no class is left behind"* criteria, where we may boost those classes that are prone to be confused in the hope of recovering more errors. Note that the design matrix could also directly encode knowledge of domain experts on the problem, providing a great flexibility on the design of the ECOC coding matrix. In addition, by using the Design matrix we derive the optimal problem-dependent code length for ECOCs obtained by means of ECF, which to the best of our knowledge is the first time this question is tackled in the extended ECOC literature. In addition, we show how ECF converges to a solution with negligible objective value when the design matrix follows certain constraints.

5.1 Error-Correcting Factorization

We propose to cast the ECOC coding matrix optimization as a Matrix Factorization problem that can be solved efficiently using a constrained coordinate descent approach. This section describes the objective function and the optimization strategy for the ECF algorithm.

5.1.1 Objective

Our goal is to find an ECOC coding matrix that encodes the properties denoted by the design matrix $\mathbf{D}$. In this sense, ECF seeks a factorization of the design matrix $\mathbf{D} \in \mathbb{R}^{k\times k}$ into a discrete ECOC matrix $\mathbf{M} \in \{-1,+1\}^{k\times l}$. This factorization is formulated as the quadratic form $\mathbf{M}\mathbf{M}^\top$ that reconstructs $\mathbf{D}$ with minimal Frobenius distance under several constraints, as shown in Equation (5.1) [1].

$$\begin{aligned} \underset{\mathbf{M}}{\text{minimize}} \quad & \|\mathbf{D} - \mathbf{M}\mathbf{M}^\top\|_F^2 && (5.1)\\ \text{subject to} \quad & \mathbf{M} \in \{-1,+1\}^{k\times l} && (5.2)\\ & \mathbf{M}\mathbf{M}^\top - \mathbf{P} \le 0 && (5.3)\\ & \mathbf{M}^\top\mathbf{M} - \mathbf{1}(l-1) \le 0 && (5.4)\\ & -\mathbf{M}^\top\mathbf{M} - \mathbf{1}(l-1) \le 0 && (5.5) \end{aligned}$$

The component $\mathbf{M} \in \{-1,+1\}^{k\times l}$ that solves this optimization problem generates the inner product of discrete vectors that is closest to $\mathbf{D}$ under the Frobenius norm. In order for $\mathbf{M}$ to be a valid matrix under the ECOC framework we constraint $\mathbf{M}$ in Equations (5.2)-(5.5). Equation (5.2) ensures that each binary problem classes will belong to one of the two possible meta-classes. In addition, to avoid the case of having two or more equivalent rows in $\mathbf{M}$, the constraints in 5.3 ensure that the correlation between rows of $\mathbf{M}$ less or equal than a certain user-defined matrix $-\mathbf{1}l \le \mathbf{P} \le \mathbf{1}l$ (recall that $\mathbf{1}$ denotes a matrix or vector of all 1s of the appropriate size when used), where $\mathbf{P}$ encodes the minimum distance between any pair of codewords. $\mathbf{P}$ is a symmetric matrix with $p_{ii} = l \ \forall i$. Thus, by setting the off diagonal values in $\mathbf{P}$ we can control the minimum inter-class correction capability. Hence, if we want the correction capability of rows $\mathbf{m}^i$ and $\mathbf{m}^j$ to be $\lfloor \frac{c-1}{2} \rfloor$, we set $\mathbf{p}^i = \mathbf{p}^j = \mathbf{1}(l-c)$.

Finally, constraints in Equations (5.4) and (5.5) ensure the induced binary problems are not equivalent. Similar constraints have been studied thoroughly in literature [39, 62, 84] defining methods that rely on diversity measures for binary problems to obtain a coding matrix $\mathbf{M}$. Equations (5.4) and (5.5) can be considered as soft-constraints since its violation does not imply violating the ECOC properties in terms of row distance. This is easy to show since a coding matrix $\mathbf{M} \in \{-1,+1\}^{k\times l}$ that induces some equivalent binary problems but ensures that $\mathbf{M}\mathbf{M}^\top \le \mathbf{1}(l-1), \ \forall i,j : i \neq j$ will define a matrix whose rows are unequivocally defined. In this sense, a coding matrix $\mathbf{M}$ can be easily projected on the set defined by constraints (5.4) and (5.5) by eliminating repeated columns, $\mathbf{M} = \mathbf{m}_j : \mathbf{m}_j \neq \mathbf{m}_i \forall j \neq i$. Thus, constraints in 5.4 and 5.5 ensure that uncorrelated binary sub-problems will be defined in our coding matrix $\mathbf{M}$. The discrete constraint in Equation 5.2 on the variable elevates the optimization problem to the NP-Hard class. To overcome this issue and following [20, 32, 142] we relax the discrete constraint in 5.2 an replace it by $\mathbf{M} \in [-1,+1]^{k\times l}$ in Equation 5.7.

5.1.2 Optimization

In this section, we detail the process for optimizing $\mathbf{M}$. The minimization problem posed in Equation (5.1) with the relaxation of the boolean constraint in Equation (5.2) is non-convex, thus, $\mathbf{M}^*$ is not guaranteed to be a global minimum. In this sense, although gradient descent techniques have been successfully applied in the literature to obtain local minimums [2, 83, 112] these techniques do not enjoy the efficiency and scalability properties present in other optimization methods applied to Matrix Factorization problems, such as Coordinate

[1] Recall that the l_1 distance is a function of the dot product $\|\mathbf{m}^i - \mathbf{m}^j\|_1 = \frac{-(\mathbf{m}^i\mathbf{m}^{j\top})+l}{2}$.

Descent [36, 88]. Coordinate Descent techniques have been widely applied in Nonnegative Matrix Factorization obtaining satisfying results in terms of efficiency [66, 79]. In addition, it has been proved that if each of the coordinate sub-problems can be solved exactly, Coordinate Descent converges to a stationary point [65, 127]. Using this result, we decouple the problem in Equation (5.1) into a set of linear least-squares problems (one for each coordinate). Therefore, if the problem in Equation (5.1) is going to be minimized along the $i-$th coordinate of $\mathbf{M}$, we fix all rows of $\mathbf{M}$ except of $\mathbf{m}^i$ and we substitute $\mathbf{M}$ with $\begin{bmatrix}\mathbf{m}^i \\ \mathbf{M}'^i\end{bmatrix}$ in Equations (5.1) and (5.3), where $\mathbf{M}'^i$ denotes matrix $\mathbf{M}$ after removing the $i-$th row. In addition, we substitute $\mathbf{D}$ with $\begin{bmatrix} l & \mathbf{d}_i \\ \mathbf{d}^{iT} & \mathbf{D}'^i_{/i}\end{bmatrix}$, where $\mathbf{D}'^i_{/i}$ denotes the matrix $\mathbf{D}$ after removing the $i-$th row and column. Equivalently, we substitute $\mathbf{P} = \begin{bmatrix} l & \mathbf{p}_i \\ \mathbf{p}^{iT} & \mathbf{P}'^i_{/i}\end{bmatrix}$, obtaining the following block decomposition:

$$\underset{\mathbf{m}^i}{\text{minimize}} \quad \left\| \begin{bmatrix} l & \mathbf{d}_i \\ \mathbf{d}^{iT} & \mathbf{D}'^i_{/i}\end{bmatrix} - \begin{bmatrix} \mathbf{m}^i\mathbf{m}^{iT} & \mathbf{M}'^i\mathbf{m}_i \\ \mathbf{M}'^i\mathbf{m}^{iT} & \mathbf{M}'^i\mathbf{M}'^{i\top}\end{bmatrix} \right\|_F^2 \tag{5.6}$$

$$\text{subject to} \quad \mathbf{m}^i \in [-1,+1]^l \tag{5.7}$$

$$\begin{bmatrix} \mathbf{m}^i\mathbf{m}^{iT} & \mathbf{M}'^i\mathbf{m}_i \\ \mathbf{M}'^i\mathbf{m}^{iT} & \mathbf{M}'^i\mathbf{M}'^{i\top}\end{bmatrix} - \begin{bmatrix} l & \mathbf{p}_i \\ \mathbf{p}^{iT} & \mathbf{P}'^i_{/i}\end{bmatrix} \leq 0. \tag{5.8}$$

Analyzing the block decomposition in Equation (5.6) we can see that the only terms involving free variables are $\mathbf{m}^i\mathbf{m}^{i\top}$, $\mathbf{M}'^i\mathbf{m}^i$ and $\mathbf{M}'^i\mathbf{m}^{i\top}$. Thus, since $\mathbf{D}$ and $\mathbf{M}\mathbf{M}^\top$ are symmetric by definition, the minimizer $\mathbf{m}^{i*}$ of Equation (5.6) is the solution to the linear least-squares problem shown in Equation (5.9):

$$\underset{\mathbf{m}^i}{\text{minimize}} \quad \left\|\mathbf{M}'^i\mathbf{m}^i - \mathbf{d}^i\right\|_2^2 \tag{5.9}$$

$$\text{subject to} \quad -1 \leq \mathbf{m}^i \leq +1 \tag{5.10}$$

$$\mathbf{M}'^i\mathbf{m}^i - \mathbf{p}^i \leq 0, \tag{5.11}$$

where constraint (5.10) is the relaxation of the discrete constraint (5.2). In addition, constraint (5.11) ensures the correlation of $\mathbf{m}^i$ with the rest of the rows of $\mathbf{M}$ is below a certain value $\mathbf{p}^i$. Algorithm 5 shows the complete optimization process.

To solve the minimization problem in Algorithm 5 we use the Active Set method described in [31], which finds an initial feasible solution by first solving a linear programming problem. Once ECF converges to a solution $\mathbf{M}^*$ with objective value $f_{obj}(\mathbf{M}^*)$ we obtain a discretized ϵ-suboptimal solution $\mathbf{M} \in \{-1,+1\}$ with objective value $f_{obj}(\mathbf{M})$ by sampling 1000 points that split the interval $[-1,+1]$ and choosing the point that minimizes $\|f_{obj}(\mathbf{M}^*) - f_{obj}(\mathbf{M})\|_2$. Finally, we discard repeated columns if any appear [2].

5.1.3 Connections to Singular Value Decomposition, Nearest Correlation Matrix and Discrete Basis problems

Similar objective functions to the one defined in the ECF problem in Equation (5.1) are found in other contexts, for example, in the Singular Value Decomposition problem (SVD).

[2] In all our runs of ECF this situation happened with a chance of less than 10^{-5}%.

```
Data: $\tilde{\mathbf{D}} \in \mathbb{R}^{k\times k}, \mathbf{P} \in \mathbb{N}^{k\times k}, l$
Result: $\mathbf{M} \in \{-1,+1\}^{k\times l}$
begin
    repeat
        foreach $i \in \{1,2,\ldots,k\}$ do
            $\mathbf{m}^i \leftarrow \underset{\mathbf{m}^i\in\mathbb{R}^l}{\text{minimize}} \left\|\mathbf{M}'^i\mathbf{m}^i - \mathbf{d}^i\right\|_2^2$, subject to : $-1 \leq \mathbf{m}^i \leq$
            $+1,\ \ \mathbf{M}'^i\mathbf{m}^i - \mathbf{p}^i \leq 0$;
        end
        $\mathbf{M} \leftarrow \epsilon$-suboptimal($\mathbf{M}$);
        $\mathbf{M} = \{\mathbf{m}_j : \mathbf{m}_j \neq \mathbf{m}_i \forall j \neq i\}$; // Projection step to remove
        duplicate columns
    until convergence;
end
```

Algorithm 5: Error-Correcting Factorization Algorithm.

The SVD uses the same objective function as ECF subjected to the constraint $\mathbf{M}\mathbf{M}^\top = \mathbf{I}$. However, the solution of SVD yields an orthogonal basis, disagreeing with the objective defined in Equation (5.1) which ensures different correlations between the $\mathbf{m}^i$'s. In addition, we can also find a common ground with the Nearest Correlation Matrix (NMC) Problem [21, 69, 93]. However, the NMC solution does not yield a discrete factor $\mathbf{M}$, instead it seeks directly for the Gramian $\mathbf{M}\mathbf{M}^\top$ where $\mathbf{M}$ is not discrete, as in Equation (5.12).

$$\begin{aligned}
&\underset{\mathbf{M}}{\text{minimize}} && \|\mathbf{M}-\mathbf{D}\|_F^2 && (5.12)\\
&\text{subject to} && \mathbf{M} \succeq 0 && (5.13)\\
& && \mathbf{c}\mathbf{M}\mathbf{c}^\top = \mathbf{b} && (5.14)
\end{aligned}$$

In addition, the ECF has similarities with the Discrete Basis Problem (DBP) [96], since the factors are $\mathbf{M}$ discrete valued. Nevertheless, DBP factorizes $\mathbf{D} \in \{0,1\}^{k\times k}$ instead of $\mathbf{D} \in \mathbb{R}^{k\times k}$, as show in Equation (5.15).

$$\begin{aligned}
&\underset{\mathbf{M},\mathbf{Y}}{\text{minimize}} && \|\mathbf{M}\circ\mathbf{Y}-\mathbf{D}\|_1 && (5.15)\\
&\text{subject to} && \mathbf{M},\mathbf{Y},\mathbf{D} \in \{0,1\} && (5.16)
\end{aligned}$$

5.1.4 Ensuring a representable design matrix

An alternative interpretation for ECF is that it seeks for a discrete matrix $\mathbf{M}$ whose Gramian is closest to $\mathbf{D}$ under the Frobenius norm. However, since $\mathbf{D}$ can be directly set by the user we need to guarantee that $\mathbf{D}$ is a correlation matrix that is realizable in the $\mathbb{R}^{k\times k}$ space, that is, $\mathbf{D}$ has to be symmetric and positive semi-definite. In particular, we would like to find the correlation matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{k\times k}$ that is closest to $\mathbf{D}$ under the Frobenius norm. This problem has been treated in several works [21, 28, 63, 69], resulting in various algorithms that often use an alternating projections approach. However, for this particular case in addition to be in the Positive Semidefinite (PSD) Cone and symmetric we also require $\mathbf{D}$ to be scaled in the $[-l,+l]$ range, with $\tilde{\delta}_{ii} = l \forall i$. In this sense, to find $\tilde{\mathbf{D}}$ we follow an alternating projections algorithm, similar as [69], which is shown in Algorithm 6. We first project $\mathbf{D}$ into the PSD cone by computing its eigenvectors and recovering $\mathbf{D} = \mathbf{V}\,\mathrm{diag}(\boldsymbol{\lambda}_+)\mathbf{V}^\top$, where $\boldsymbol{\lambda}_+$ are the non-negative eigenvalues of $\mathbf{D}$. Then, we scale $\mathbf{D}$ in the range $[-l, +l]$ and set $\delta_{ii} = l \forall i$.

```
Data: $\mathbf{D} \in \mathbb{R}^{k \times k}$
Result: $\tilde{\mathbf{D}} \in \mathbb{R}^{k \times k}$
begin
    repeat
        $\mathbf{D} \leftarrow \mathbf{V}\,\mathrm{diag}(\boldsymbol{\lambda}_+)\mathbf{V}^\top$;
        $\mathbf{D} \leftarrow \mathbf{D} \in [-l, +l]^{k \times k}$;
        $\mathbf{D} \leftarrow d_{ii} = l \forall i$;
    until convergence;
end
```

Algorithm 6: Projecting $\mathbf{D}$ into the PSD cone with additional constraints.

5.1.5 Defining a code length with representation guarantees

The definition of a problem-dependent ECOC code length l, that is, choosing the number of binary partitions for a given multi-class task is a problem that has been overlooked in literature. For example, predefined coding designs like One vs. All or One vs. One have fixed code length. On the other hand, coding designs like Dense or Sparse Random codings (which are very often used in experimental comparisons [7, 51, 141, 142]) are suggested [3] to have a code length of $\lceil 10 log_2(k) \rceil$ and $\lceil 15 log_2(k) \rceil$ respectively. These values are arbitrary and unjustified. Additionally, to build a Dense or Sparse Random ECOC matrix one has to generate a set of 1000 matrices and chose the one that maximizes $\min(\mathbf{H})$. Consider the Dense Random Coding design, of length $l = \lceil 10 \log_2(k) \rceil$, the ECOC matrix will have in the best case a correction capability of $\lfloor \frac{10-1}{2} \rfloor = 4$, independently of the distribution of the multi-class data. In addition, the effect of maximizing $\min(\mathbf{H})$ leads to an equi-distribution of the correction capability over the classes. Other approaches, like Spectral ECOC [141] search for the code length by looking at the best performance on a validation set. Nevertheless, recent works have shown that the code length can be reduced to of $l = log_2(k)$ with very small loss in performance if the ECOC coding design is carefully chosen [89] and classifiers are strong. In this paper, instead of fixing the code length or optimizing it on a validation subset, we derive the optimal length according to matrix rank properties. Consider the rank of a factorization of $\mathbf{D}$ into $\mathbf{M}\mathbf{M}^\top$, there are three different possibilities:

1. If $\mathrm{rank}(\mathbf{M}\mathbf{M}^\top) = \mathrm{rank}(\mathbf{D})$, we obtain rank factorization algorithm that should be able to factorize $\mathbf{D}$ with minimal error.
2. In the case when $\mathrm{rank}(\mathbf{M}\mathbf{M}^\top) < \mathrm{rank}(\mathbf{D})$ we obtain a low-rank factorization method that cannot guarantee to represent $\mathbf{D}$ with 0 error, but reconstructs the components of $\mathbf{D}$ with higher information.
3. If $\mathrm{rank}(\mathbf{M}\mathbf{M}^\top) > \mathrm{rank}(\mathbf{D})$, the system is overdetermined and many possible solutions exist.

In general we would like to reconstruct $\mathbf{D}$ with minimal error, and since $\mathrm{rank}(\mathbf{M}) \leq \min(k, l)$ and k (the number of classes) is fixed, we only have to set the number of columns of $\mathbf{M}$ to control the rank. Hence, by setting $\mathrm{rank}(\mathbf{M}) = l = \mathrm{rank}(\mathbf{D})$, ECF will be able to factorize $\mathbf{D}$ with minimal error. Figure 5.1 shows visual results for the ECF method applied

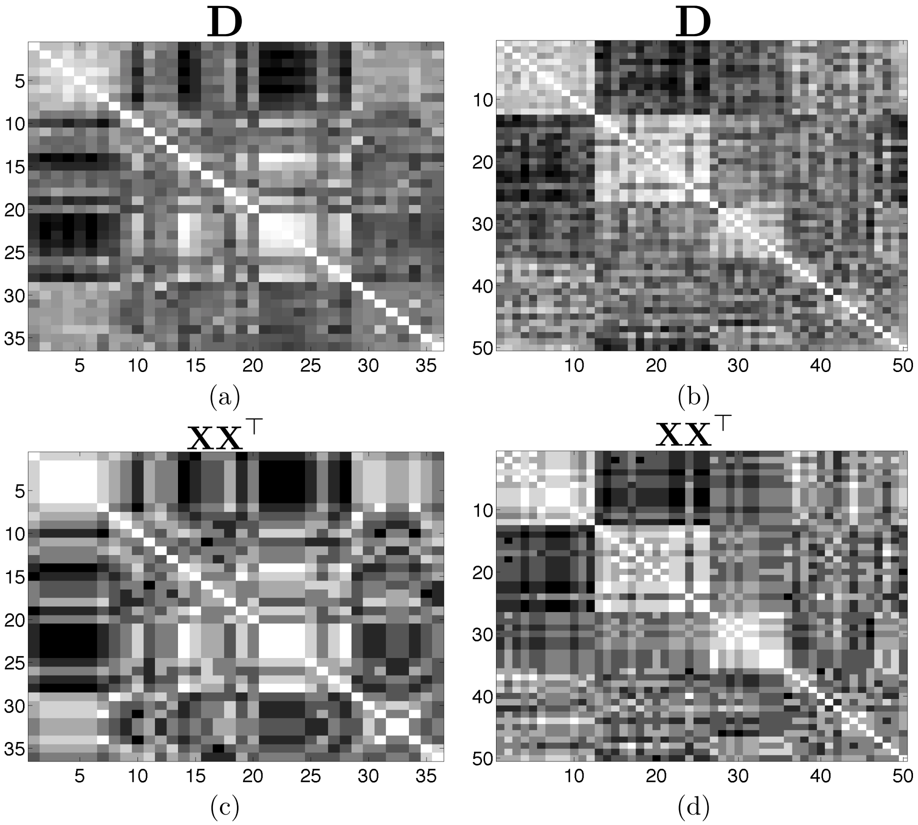


Figure 5.1: D matrix for the *Traffic* (a) and *ARFace* (b) datasets. $\mathbf{MM}^\top$ term obtained via ECF for *Traffic* (c) and *ARFace* (d) datasets. ECOC coding matrix **M** obtained with ECF for *Traffic* (e) and *ARFace* (f).

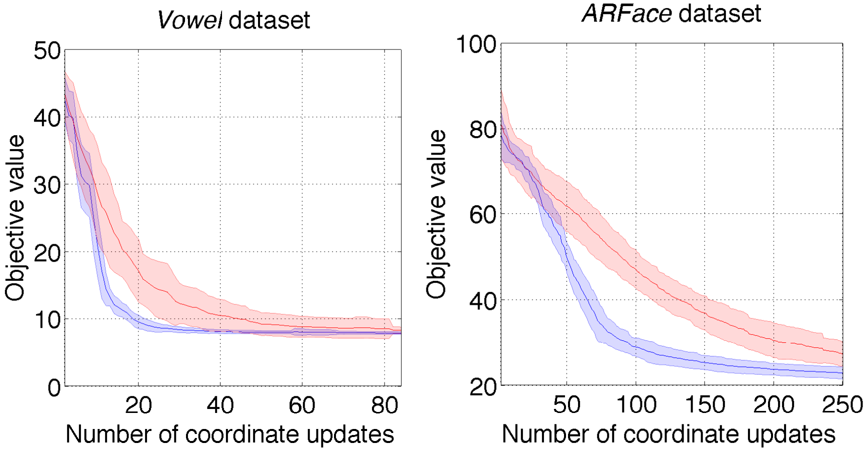
on the *Traffic* and *ARFace* datasets. Note how, for the *Traffic* (36 classes) and *ARFaces* (50 classes) datasets the required code length for ECF to full rank factorization is $l = 6$ and $l = 8$, respectively as shown in Figures 5.1(e)(f).

5.1.6 Order of Coordinate Updates

Coordinate Descent has been applied in a wide span of problems obtaining satisfying results. However, the problem of choosing the coordinate to minimize at each iteration still remains active [58, 71, 110, 127]. In particular, [99] derives a convergence rate which is faster when coordinates are chosen uniformly at random rather than on a cyclic fashion. Hence, choosing coordinates at random its a suitable choice when the problem shows some of the following characteristics [110]:

- Not all data is available at all times.
- A randomized strategy is able to avoid worst-case order of coordinates, and hence might be preferable.
- Recent efforts suggest that randomization can improve the convergence rate [99].

However, the structure of ECF is different and calls for a different analysis. In particular, we remark the following points. (i) At each coordinate update of ECF, information about the rest of coordinates is available. (ii) Since our coordinate updates are solved uniquely, a repetition on a coordinate update does not change the objective function. (iii) The descent on the objective value when updating a coordinate is maximal when all other coordinates have been updated. These reasons leads us to choose a cyclic update scheme for ECF. In addition in Figure 5.2 we show a couple of examples in which the cyclic order of coordinates converges faster than the random order for two problems: *Vowel* and *ARFace* (refer to Section 5.2 for further information on the datasets). This behavior is common for all datasets. In particular, note how the cyclic order of coordinates reduces the standard deviation on the objective function, which is denoted by the narrower blue shaded area in Figure 5.2.



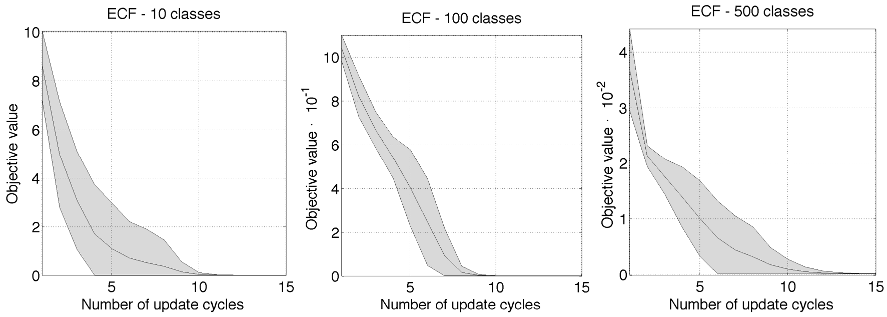
(a)

(b)

Figure 5.2: Mean Frobenius norm value with standard deviation as a function of the number of coordinate updates on 50 different trials. The blue shaded area corresponds to cyclic update while the red area denotes random coordinate updates for *Vowel* (a) and *ARFAce* (b) datasets.

5.1.7 Approximation Errors and Convergence results when D is an inner product of binary data

The optimization problem posed by ECF in Equation (5.1) is non-convex due to the quadratic term $\mathbf{M}\mathbf{M}^\top$, even if the discrete constraint is relaxed. This implies that we cannot guarantee that the algorithm converges to the global optima. Recall that ECF seeks for the term $\mathbf{M}\mathbf{M}^\top$ that is closest to $\mathbf{D}$ under the Frobenius norm. Hence, the error in the approximation can be measured by $\|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}\|_F^2 \geq 0$, where $\mathbf{M}^*$ is the local optimal point to which ECF converges. In this sense, we introduce $\mathbf{D}^B$ which is the matrix of inner products of discrete vectors that is closest to $\mathbf{D}$ under the Frobenious norm. Thus, we expand the norm as in the following equation:



(a) (b) (c)

Figure 5.3: (Mean objective value and standard deviation for 30 runs of ECF on a random $\mathbf{D}^G$ of 10 classes (a), 100 classes (b), and 500 classes (c). (d) Toy problem synthetic data, where each color corresponds to a different category in the multi-class problem.

$$\begin{aligned}
\|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}\|_F^2 &= \|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}^B + \mathbf{D}^B - \mathbf{D}\|_F^2 = &(5.17)\\
&= \|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}^B\ \|_F^2 + \|\mathbf{D} - \mathbf{D}^B\ \|_F^2 - &(5.18)\\
&\quad -2\,\mathrm{tr}((\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}^B)(\mathbf{D} - \mathbf{D}^B)). &(5.19)
\end{aligned}$$

- The optimization error ε_o: measured as the distance between the local optimum where ECF converges and $\mathbf{D}^B$ denoted by $\varepsilon_o = \|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}^B\|_F^2$, which is expressed as the first term in Equation (5.18).

- The discretization error ε_d: computed as, $\varepsilon_d = \|\mathbf{D} - \mathbf{D}^B\|_F^2$, that is, the distance between $\mathbf{D}$ and the closest inner product of discrete vectors $\mathbf{D}^B$, expressed as the second term in Equation (5.18).

In order to better understand how ECF works we analyze both components separately. Then, to analyze if ECF converges to a good solution in terms of Frobenius norm we set $\varepsilon_d = 0$ by generating a matrix $\mathbf{D} = \mathbf{D}^B$ which is the inner product matrix of random discrete vectors, and thus, all the terms except of $\|\mathbf{M}^*\mathbf{M}^{*\top} - \mathbf{D}^B\|_F^2$ are zero. By doing that, we can empirically observe the magnitude of the optimization error ε_o. In order to do that we run ECF 30 times on 100 different $\mathbf{D}^B$ matrices of different sizes and calculate the average $\bar{\varepsilon_o}$. Figure 5.3 shows examples for different $\mathbf{D}^G$ matrices of size 10×10, 100×100, and 500×500. In Figure 5.3 we can see how ECF converges to a solution with almost negligible optimization error after 15 iterations. In fact, the average objective value for all 3000 runs of ECF on different $\mathbf{D}^B$'s after 15 update cycles (coordinate updates for all $\mathbf{m}^i$'s) is $\bar{\varepsilon_o} < 10^{-10}$. This implies, that ECF converges in average to a point with almost negligible objective value, and when applied to $\mathbf{D}$'s which are not computed from binary components the main source of the approximation error is the discretization error ε_d. Since ECF seeks to find a discrete decomposition of $\mathbf{D}$ this discretization error is unavoidable, and as we have seen empirically, ECF converges in average to a solution with almost negligible objective value.

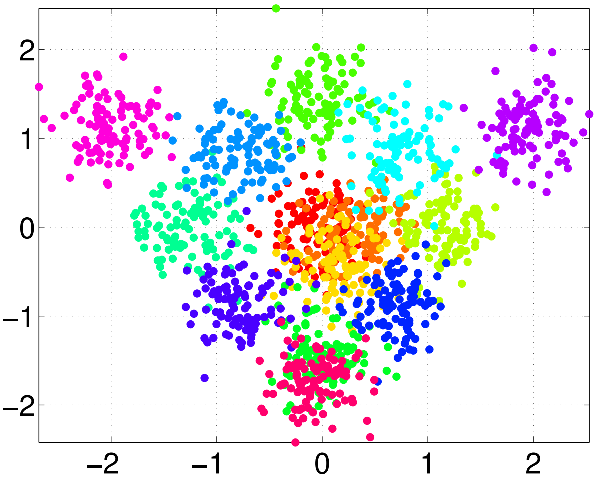


Figure 5.4: Toy problem synthetic data, where each color corresponds to a different category in the multi-class problem.

5.2 Experiments

In this section we present the experimental results of the proposed Error-Correcting Factorization method. In order to do so, we first present the data, methods and settings.

5.2.1 Data

The proposed Error-Correcting Factorization method was applied to a total of 8 datasets. In order to provide a deep analysis and understanding of the method, we synthetically generated a *Toy* problem consisting of $k = 14$ classes, where each class contained 100 two dimensional points sampled from a Gaussian distribution with same standard deviation but different means. Figure 5.4(d) shows the synthetic multi-class generated data, where each color corresponds to a different category. We selected 5 well-known UCI datasets: *Glass*, *Segmentation*, *Ecoli*, *Yeast* and *Vowel* that range in complexity and number of classes. Finally, we apply the classification methodology in two challenging computer vision categorization problems. First, we test the methods in a real traffic sign categorization problem consisting of 36 traffic sign classes. Second, 50 classes from the ARFaces [95] dataset are classified using the present methodology. These datasets are public upon request to the authors. Detailed descriptions and characteristics for each dataset can be found in Appendix A.

5.2.2 Methods and settings

We compared the proposed Error-Correcting Factorization method, with the standard predefined One vs. All (*OVA*) and One vs. One (*OVO*) approaches [111, 125]. In addition, we introduce two random designs for ECOC matrices. In the first one, we generated random ECOC coding matrices fixing the general correction capability to a certain value (*RAND*). In the second, we generate a Dense Random coding matrix [4] (*DENSE*). These comparisons enable us to analyze the effect of reorganizing the inter-class correcting capabilities of an ECOC matrix. Finally, in order to compare our proposal with state-of-the-art methods, we also used the Spectral ECOC (*S-ECOC*) method [141] and the Relaxed Hierarchy [60] (*R-H*) . Finally we propose two different flavors of ECF, *ECF-H* and *ECF-E*. In *ECF-H* we compute the design matrix **D** in order to allocate the correction capabilities on those classes that are **hard** to discriminate. On the other hand, for *ECF-E* we compute **D** allocating correction to those classes that are **easy** to discriminate. **D** is computed as the Mahalanobis distance between each pair of classes. Although, there exist a number of approaches to define **D** from data [60, 141, 142], i.e. the margin between each pair of classes (after training a *One vs. One* SVM classifier), we experimentally observed that the Mahalanobis distance provides good generalization and leverages the computational cost of training a *One vs. One* SVM classifier. All the reported classification accuracies are the mean of a stratified $5-$fold cross-validation on the aforementioned datasets. For all methods we used an SVM classifier with RBF kernel. The parameters C and γ were tunned by cross-validation on a validation subset of the data using an inner $2-$fold cross-validation. The parameter C was tunned on a grid-search on a log sampling in the range $[0, 10^{10}]$, and the γ parameter was equivalently tuned on a equidistant linear sampling in the range $[0, 1]$, we used the libsvm implementation available at [29]. For both *ECF-H* and *ECF-E* we run the factorization forcing different minimum distance between classes by setting $\mathbf{P} \in \mathbf{1} \cdot \{1, 3, 5, 7, 10\}$. For the Relaxed Hierarchy method [60] we used values for $\rho \in \{0.1, 0.4, 0.7, 0.9\}$. In all the compared methods that use a decoding function (e.g all tested methods but the one in [60]) we used both the Hamming Decoding (HD) and the Loss-Weighted decoding (LWD) [106].

5.2.3 Experimental Results

In Figure 5.5 we show the multi-class classification accuracy as a function of the relative computational complexity for all datasets using both Hamming decoding (HD) and Loss-Weighted Decoding (LWD). We used non-linear SVM classifiers and we define the relative computational complexity as the number of unique Support Vectors (SVs) yielded for each method, as in [60]. For visualization purposes we use an exponential scale and normalize the number of SVs by the maximum number of SVs obtained by a method in that particular dataset. In addition, although the code length cannot be considered as an accurate measure of complexity when using non-linear classifiers in the feature space, it is the only measure of complexity that is available prior to learning the binary problems and designing the coding matrix. In this sense, we show in Figure 5.6 the classification results for all datasets as a function of the code length l, using both Hamming decoding (HD) and Loss-Weighted Decoding (LWD). Figures 5.5 and 5.6 and show how the proposed *ECF-H* obtains in most of the cases better performance than state-of-the-art approaches even with reduced computational complexity. In addition, in most datasets the *ECF-H* is able to boost the boundaries of those classes prone to error, the effect of this is that it attains higher classification accuracies than the rest of methods paying the prize of an small increase on the relative computational complexity. Specifically, we can see how on *Glass* dataset, *Vowel*, *Yeast*, *Segmentation* and *Traffic* datasets (Figs. 5.5(e)-(f) and 5.6(e)-(f), respectively), the proposed method outperforms the rest of the approaches while yielding a comparable or even lower computational complexity, independently of the decoding function used. We also can see that the *RAND* and *ECF-E* methods present erratic behaviours. This is expected for the random coding design, since incrementing the number of SVs or dichotomies does not imply an increase in performance if the dichotomies are not carefully selected. On the other hand, the reason why *ECF-E* is not stable is not completely straightforward. *ECF-E* focus its design in dichotomies that are very easy to learn, allocating correction to those classes that are separable. We hypothesize that when these dichotomies become harder (there exists a limited number of easy separable partitions) to learn the addition of a difficult dichotomy harms the performance by adding

Table 5.1: Percentage of wins over all datasets for each method using as a complexity measure the number SVs and the number of classifiers. Last row shows the average complexity of each method over all datasets. Abbreviations: ECF-H (H), ECF-E (E), OVA (A), OVO (O), DENSE (D), RAND (R), S-ECOC(S).

Method	*R-H**	*S*	*H*	*E*	*D*	*R*	*A*	*O*
Win % SVs	0.0	22.5	**62.1**	10.3	50.0	5.7	14.2	25.0
Win % nclass.	0.0	48.5	**70.0**	17.5	25.0	6.9	12.5	16.6
Avg. Comp.	**0.58**	0.87	0.88	0.89	0.91	0.92	0.99	0.99

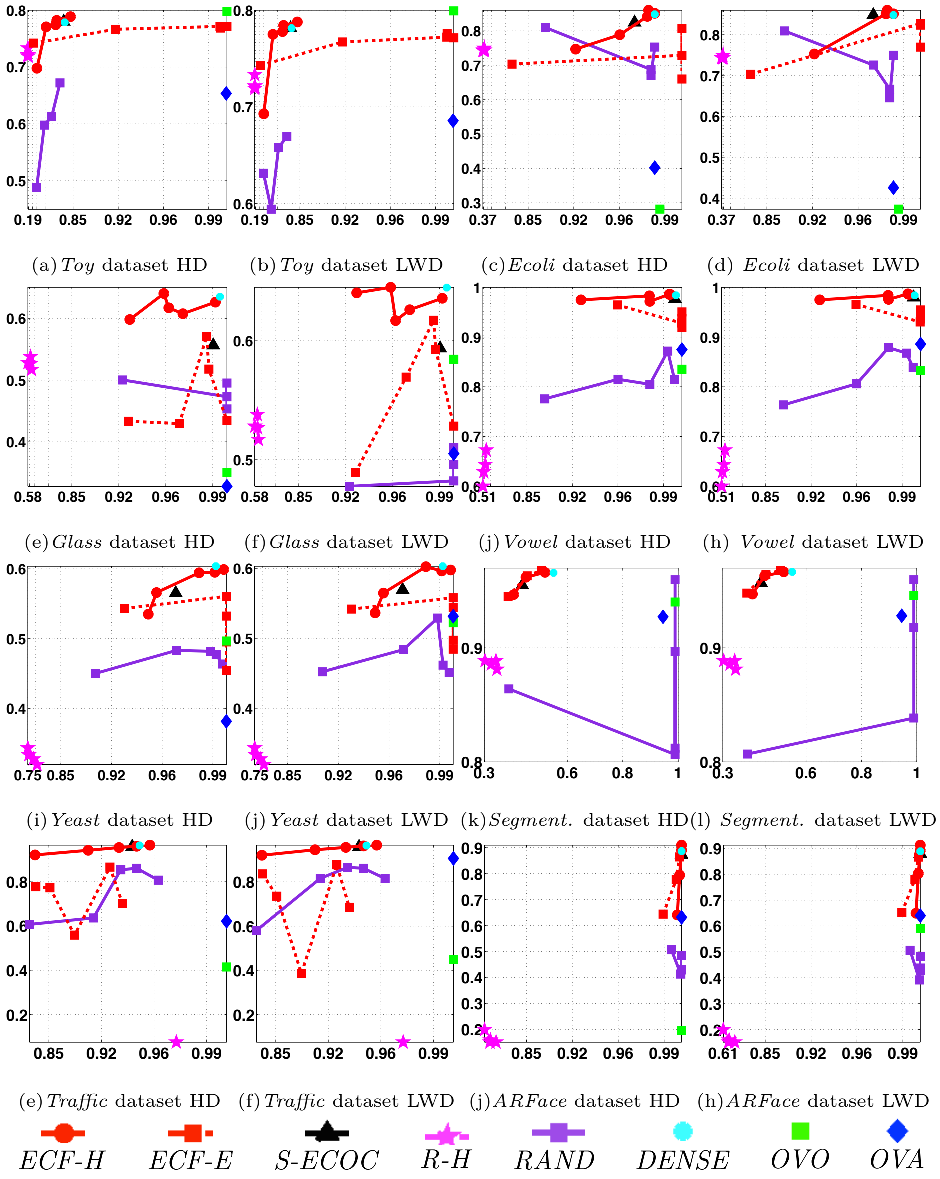
confusion to previously learned dichotomies until proper error-correction is allocated. On the other hand, we can see how *ECF-H* usually shows a more stable behaviour since it focuses on categories that are prone to be confused. In this sense, we expect that the addition of dichotomies will increase the correction. Finally, it is worth noting that the Spectral ECOC method yields a code length of $l = k - 1$, corresponding to the full eigendecomposition. Our proposal defines coding matrices which ensure to follow the design denoted by $\mathbf{D}$, fulfilling ECOC properties.

As a summary, we show in Figure 5.7 a comparison in terms of classification accuracy for different methods over all datasets. We compare the classification accuracy of a selected method for both decodings (at different operating complexities if available) versus the best performing method in a range of ±5% of the operative complexity. For consistency we show the comparison using both the number of SVs and the number of dichotomies as the computational complexity. If the compared method dominates in most of the datasets it will be found above the diagonal. In Figures 5.7(a) and 5.7(d) we compare *ECF-H* with the best performant of the rest of the methods and see that *ECF-H* outperforms the rest of the methods $62\% - 70\%$ of the times depending on the complexity measure. This implies that *ECF-H* dominates most of the methods in terms of performance by focusing on those classes that are more prone to error regardless of the complexity measure used (number of SVs or number of dichotomies). In addition, when repeating the comparison for *ECF-E* in Figures 5.7(b) and 5.7(e) we see that the majority of the datasets are clearly below the diagonal (*ECF-E* is the most suitable choice $10\% - 17\%$ of times). Finally, Figures 5.7(c) and 5.7(f) show the comparison for *OVA*, which is a standard method often defended by its simplicity [111]. We clearly see how it never outperforms any method and it is not the recommended choice for almost any dataset. In Table 5.1 we show the percentage of wins for all methods[3], in increasing order of complexity averaged over all datasets. Note how, *ECF-H* denoted by *H* in the table although being the third less complex method outperforms by far the rest of the methods with an improvement of at least $12\% - 20\%$ in the worst case. In conclusion, the experimental results show that *ECF-H* yields ECOC coding matrices which obtain comparable or even better results than state-of-the-art methods with similar relative complexity. Furthermore, by a allowing a small increase in the computational complexity when compared to state-of-the-art methods, ECF is able to obtain better classification results by boosting the boundaries of classes that are prone to be confused.

5.3 Conclusions

In this chapter we presented the Error-Correcting Factorization method for multi-class learning, which is based on the Error-Correcting Output Codes framework. The proposed method

[3] The R-H method [60] is far less complex than the compared methods, however we compare it to the to the closest operating complexity for each of the rest of the methods.



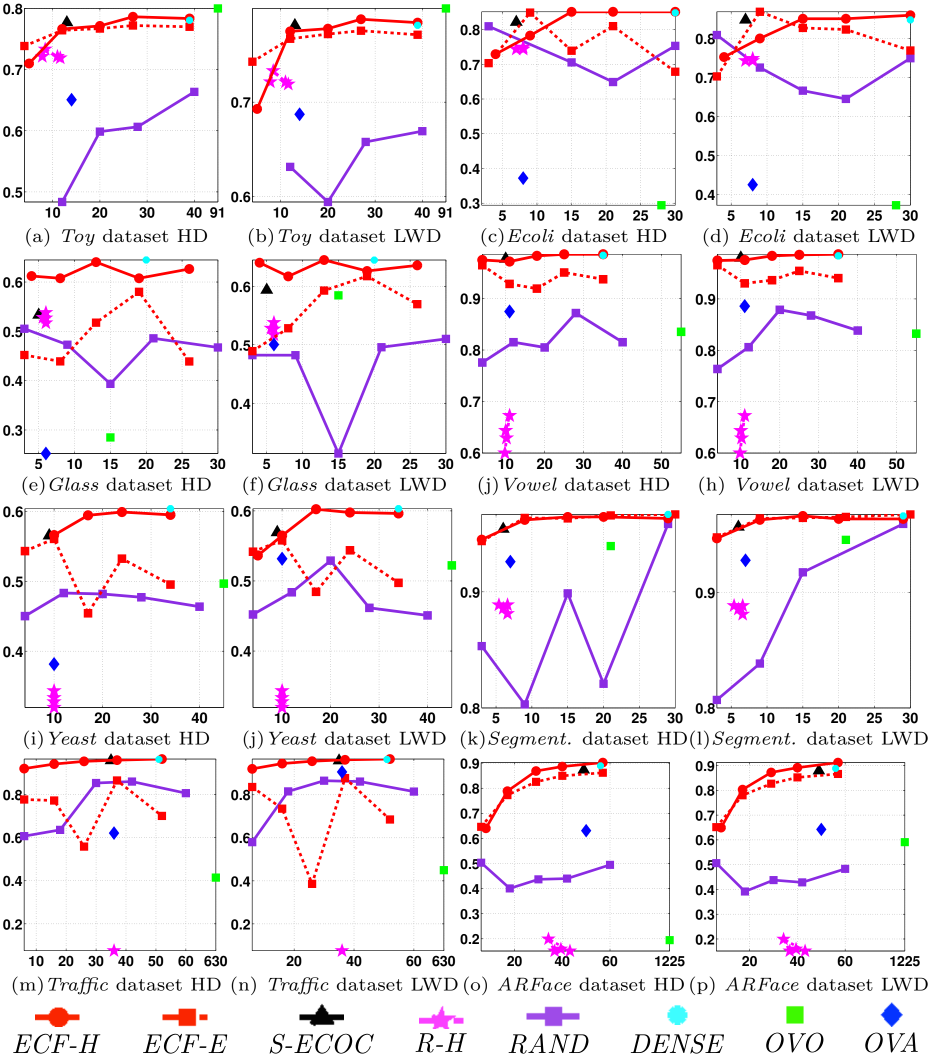
(a) *Toy* dataset HD (b) *Toy* dataset LWD (c) *Ecoli* dataset HD (d) *Ecoli* dataset LWD

(e) *Glass* dataset HD (f) *Glass* dataset LWD (j) *Vowel* dataset HD (h) *Vowel* dataset LWD

(i) *Yeast* dataset HD (j) *Yeast* dataset LWD (k) *Segment.* dataset HD (l) *Segment.* dataset LWD

(e) *Traffic* dataset HD (f) *Traffic* dataset LWD (j) *ARFace* dataset HD (h) *ARFace* dataset LWD

Figure 5.5: Multi-class classification accuracy (y axis) as a function of the relative computational complexity (x axis) for all datasets and both decoding measures.



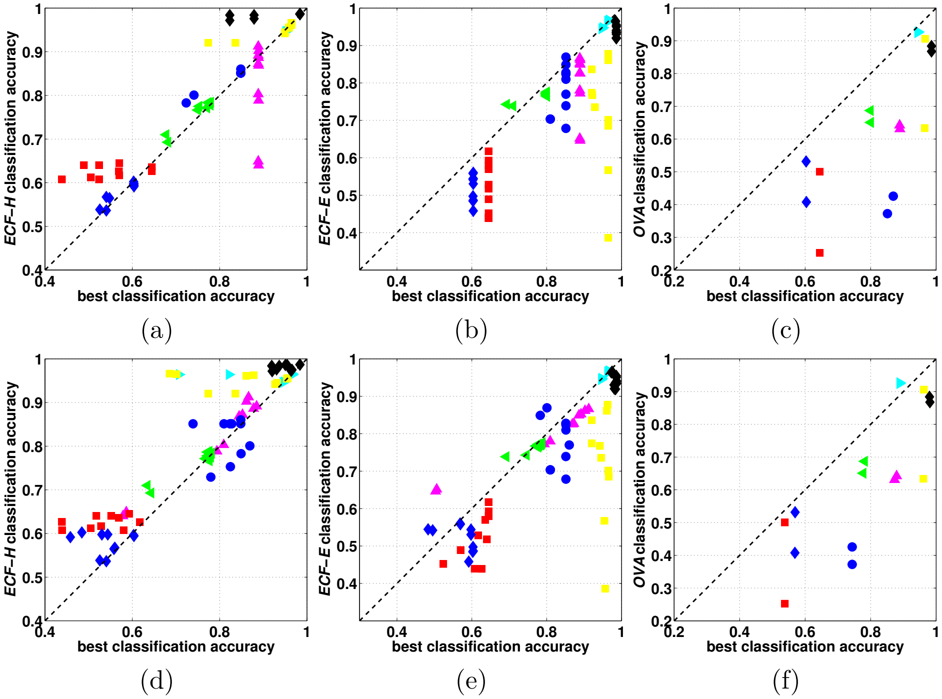
(a) *Toy* dataset HD (b) *Toy* dataset LWD (c) *Ecoli* dataset HD (d) *Ecoli* dataset LWD

(e) *Glass* dataset HD (f) *Glass* dataset LWD (j) *Vowel* dataset HD (h) *Vowel* dataset LWD

(i) *Yeast* dataset HD (j) *Yeast* dataset LWD (k) *Segment.* dataset HD (l) *Segment.* dataset LWD

(m) *Traffic* dataset HD (n) *Traffic* dataset LWD (o) *ARFace* dataset HD (p) *ARFace* dataset LWD

Figure 5.6: Multi-class classification accuracy (y axis) as a function of the number of dichotomies for all datasets and both decoding measures (x axis).



(a) (b) (c)

(d) (e) (f)

Figure 5.7: (a) Summary of performance of *ECF-H* method over all datasets using the number of SVs and the number of dichotomies as the measure of complexity, respectively for *ECF-H* (a)(d), *ECF-E* (b)(e) and *OVA* (c)(f).

factorizes a design matrix of desired correction properties into a discrete Error-Correcting component consistent with the design matrix. ECF is a general method for building an ECOC multi-class classifier with desired properties, which can be either directly set by the user or obtained from data using a priori inter-class distances. We note that the proposed approach is not a replacement for ECOC codings, but a generalized framework to build ECOC matrices that follow a certain error-correcting criterion design. The Error-Correcting Factorization is formulated as a minimization problem which is optimized using a constrained Coordinate Descent, where the minimizer of each coordinate is the solution to a least-squares problem with box and linear constraints that can be efficiently solved. By analyzing the approximation error, we empirically show that although ECF is a non-convex optimization problem, the optimization is very efficient. We performed experiments using ECF to build ECOC matrices following the common trend in state-of-the-art works, in which the design matrix priorized the most separable classes. In addition, we hypothesized and showed that a more beneficial situation is to allocate the correction capability of the ECOC to those categories which are more prone to confusion. Experiments show that when ECF is used to allocate the correction capabilities to those classes which are prone to confusion we obtain higher accuracies than state of the art methods with efficient models in terms of the number of Support Vectors and dichotomies.

Finally, there still exists open questions that require a deeper analysis for future work. The results obtained raise a fair doubt regarding the right allocation of error correcting power in several methods found in literature where ECOC designs are based on the premise of boosting the classes which are easily separable. In the light of these results, we may conjecture that a careful allocation of error correction must be made in such a way that balances two aspects: on one hand, simple to classify boundaries must be handled properly. On the other hand, the error correction must be allocated on difficult classes for the ensemble to correct possible mistakes. In addition, it would be interesting to study which are the parameters that affect the suitability of the *no class is left behind* and the *hard classes are left behind* one. Finally we could consider ternary matrices and further regularizations.

Part III

Applications in Pose Estimation Problems

Chapter 6

Applications of ECOCs in Pose Estimation Problems

The problem of recovering the pose of human beings in images has always attracted a lot of attention, due to the huge amount of possible applications in fields like Human-Computer Interaction, Surveillance, eHealth, or Gaming. Nevertheless, human multi-limb segmentation is a very hard task because of dramatic potential changes in appearance produced by different points of view, clothing, lighting conditions, occlusions, and number of articulations of the human body. From the perspective of Machine Learning the task of estimating the pose of the human body in images is often modeled as a multi-class classification problem, in which for each pixel in the image a label denoting a human limb (or background) has to be estimated. Hence, rendering this problem suitable for the Error-Correcting Output Codes framework.

In this chapter we present our contributions in human pose estimation by using ECOCs. In particular we propose a two-stage approach for the segmentation of human limbs, based on the ECOC framework and Graph-Cuts optimization. Furthermore, we also introduce the *HuPBA 8k+* dataset, which is one the biggest datasets in for human pose estimation in literature. The dataset contains more than 8000 labeled frames at pixel precision, including more than 120000 manually labeled samples of 14 different limbs.

6.1 Human Pose Estimation

The common pipeline for human pose estimation in visual data is usually defined in a bottom-up fashion. First, the human body limbs are segmented and the body pose is estimated (often with a prior person/background segmentation or person detection step). Then, once the body pose is estimated higher abstraction analysis can be performed using the predicted skeletal model.

The first step of the pipeline, which concerns pose estimation in RGB images has been a core problem in the Computer Vision field since its early beginnings. In this particular problem the goal is to provide a label for each pixel in the image whether it belongs to a certain body limb or to the background class, discriminating human limbs from each other and from the rest of the image. Usually, human body segmentation is treated in a two-stage fashion. First, a human body part detection step is performed, and then, these human part detections are used as a prior knowledge to be optimized by segmentation/inference strategies in order to encode the natural relations of the different limbs and obtain the final human-limb semantic segmentation. In literature one can find many works that follow this two-stage scheme. Bourdev et. al. [19] used body part detections in an AND-OR graph to obtain the pose estimation. Vinet et. al [130] proposed to use Conditional Random Fields based on body part detectors to obtain a complete person/background segmentation. Nevertheless, one of the methods that have generated more attraction is the well known pictorial structure for object recognition introduced by Felzenszwalb et al [54]. Some works have applied an adaptation of pictorial structures using a set of joint limb marks to infer spatial probabilities [6, 108, 109, 118]. Later on, an extension was presented by Yang and Ramanan [138, 139] which proposed a discriminatively trained pictorial structure that models the body joints instead of limbs. In contrast, there is also current tendency to use Graph Cuts optimization to segment the human limbs [68] or full person segmentation [114].

The Computer Vision community has been lately focusing their efforts on developing methods for both pose estimation and action/gesture recognition. However, one of the main problems is the necessity of public available data sets containing annotations of all the variabilities the methods have to deal with. Substantial effort has been put on designing datasets with different scenarios, people and illumination characteristics. Datasets such as Parse [108], Buffy [55], UIUC People [126], and Pascal VOC [53] are widely used to evaluate different pose estimation and action/gesture recognition methods. However, these public available datasets fail to provide with a sound framework in which to validate pose recovery systems (i.e. the number of samples per limb is small, the labeling is not accurate, there are no interactions of actors, etc.). Given this lack of sound and refined public datasets for human multi-limb segmentation and/or action/gesture recognition, in this thesis we introduce the *HuPBA 8k+* dataset, which to the best of our knowledge is the biggest RGB human-limb labeled dataset. The dataset contains more than 8000 labeled frames at pixel precision and more than 120000 manually labeled samples of 14 different limbs.

Furthermore, in this chapter of the thesis we also propose a two-stage approach for the segmentation of human limbs. In a first stage, a set of human limbs are normalized by main orientation to be rotation invariant, described using Haar-like features, and trained using cascades of Adaboost classifiers to be split in a tree-structure way. Once the tree-structure is trained, it is included in a ternary ECOC framework. This first classification step is applied in a windowing way on a new test image, defining a body-like probability map, which is used as an initialization of a binary Graph Cuts optimization procedure. In the second stage, we embed a similar tree-structured partition of limbs in a ternary ECOC framework and we use Support Vector Machines (SVMs) with HOG descriptors to build a more accurate set of limb-like probability maps within the segmented user binary mask, that are fed to a multi-label GraphCut optimization procedure to obtain the final human multi-limb segmentation. We tested our ECOC-Graph-Cut based approach in the novel *HuPBA 8k+* dataset and compared with state-of-the-art pose recovery approaches, obtaining performance improvements. Summarizing, our key contributions for this chapter are:

- We introduce the *HuPBA 8k+* dataset, the largest RGB labeled dataset of human limbs, with more than 120000 manually annotated limbs.
- We propose a two stage approach based on ECOC and Graph Cuts for the segmentation of human limbs in RGB images.
- The proposed method is compared with state-of-the-art methods for human pose estimation obtaining very satisfying results.

6.2 HuPBA 8K+ Dataset

Human pose recovery is a challenging problem in Computer Vision, not only for the intrinsic complexity of the task, but also for the lack of large public and annotated datasets. Usually, public available datasets lack of refined labeling or contain a very reduced number of samples per limb (e.g. *Buffy Stickmen V*3.01, *Leeds Sports* and *Hollywood Human Actions* [55, 77, 85]). In addition, large datasets often use synthetic samples or capture human limbs with sensor technologies such as *MoCap* in very controlled environments [37].

Being aware of this lack of public available datasets for multi-limb human pose detection, segmentation and action/gesture recognition, we present a novel fully limb labeled dataset, the *HuPBA* 8*k*+ dataset. This dataset is formed by more than 8000 frames where 14 limbs are labeled at pixel precision[1].

1. The images are obtained from 9 videos (RGB sequences) and a total of 14 different actors appear in those 9 sequences. In concrete, each sequence has a main actor (9 in total) which during the video interacts with secondary actors performing a set of different actions.
2. RGB images were stored with resolution 480 × 360 in BMP file format.
3. For each actor present in an image 14 limbs (if not occluded) were manually tagged: Head, Torso, R-L Upper-arm, R-L Lower-arm, R-L Hand, R-L Upper-leg, R-L Lower-leg, and R-L Foot.
4. Limbs are manually labeled using binary masks and the minimum bounding box containing each subject is defined.
5. The actors appear in a wide range of different poses.

Sample of different poses in the HuPBA 8*k*+ dataset are show in Figure 6.1. Further details of the dataset can be found in [117].

Finally, in Table 6.1 we compare the HuPBA 8*k*+ dataset characteristics with some publicly available datasets. These public datasets are chosen taking into account the variability of limbs and gestures/actions. Once can see that the novel dataset offers higher number of annotated limbs at pixel precision in comparison with state-of-the-art public available datasets.

	HuPBA	PARSE[108]	BUFFY[55]	UIUC[126]	LEEDS[77]	HW[85]	MMGR13[46]	H.Actions[120]	Pascal VOC[53]
Labeling at pixel precision	Yes	No	No	No	No	-	No	No	Yes
Number of limbs	14	10	6	14	14	-	16	-	5
Number of labeled limbs	124 761	3 050	4 488	18 186	28 000	-	27 532 800	-	8 500
Number of frames	8 234	305	748	1 299	2 000	-	1 720 800	-	1 218
Full body	Yes	Yes	No	Yes	Yes	-	Yes	Yes	Yes
Limb annotation	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Gesture annotation	Yes	No	No	No	No	Yes	Yes	Yes	No
Number of gestures	11	-	-	-	-	8	20	6	-
Number of gesture samples	235	-	-	-	-	430	13 858	600	-

Table 6.1: Comparison of public dataset characteristics.

[1] The whole number of manual labeled limbs exceeds 120000.

6.3 ECOC and GraphCut based multi-limb segmentation

In the following subsections we describe the proposed ECOC-based method for automatic segmentation of human limbs. To accomplish this task, we define a framework divided in two stages. The first stage, focused on binary person/background segmentation, is split in four main steps: a) Body part learning using cascade of classifiers, b) Tree-structure learning of human limbs, c) ECOC multi-limb detection, and d) Binary GrabCut optimization for foreground extraction. In the second stage, we segment the person/background binary mask into different limb regions. This stage is split in the following four steps: e) Tree-structure body part learning without background, f) ECOC multi-limb detection, g) Limb-like probability map definition, and h) Alpha-beta swap Graph Cuts multi-limb segmentation. The scheme of the proposed system is illustrated in Fig. 6.2.

6.3.1 Body part learning using cascade of classifiers

The core of most human body segmentation methods in the literature relies on body part detectors. In this sense, most part detectors in literature follow a cascade of classifiers architecture [30, 43, 57, 97, 148]. Cascades of classifiers are based on the idea of learning and unbalanced binary problem by using the negative outputs of a classifier f^i as an input for the following classifier f^{i+1}. Particularly, this cascade structure allows any classifier to refine the prediction by reducing the false positive rate at every stage of the cascade. In this sense, we use AdaBoost as the base classifier in our cascade architecture. In addition, in order to make the body part detection rotation invariant, all body parts are rotated to the dominant gradient region orientation. Then, Haar-like features are used to describe the body parts.

Because of its properties, cascade of classifiers are usually trained to split one visual object from the rest of possible objects of an image. This means that the cascade of classifiers learns to detect a certain object (body part in our case), ignoring all other objects (all other body parts). However, if we define our problem as a multi-limb detection procedure, some body parts are similar in appearance, and thus, it makes sense to group them in the same visual category. Because of this reason, we propose to learn a set of cascade of classifiers where a subset of limbs are included in the positive set of a cascade, and the remaining limbs are included as negative instances together with background images in the negative set of the cascade. Applying this grouping for different cascades of classifiers in a tree-structure way and combining them in an Error-Correcting Output Codes (ECOC) framework enables the system to perform multi-limb detection [50].

6.3.2 Tree-structure learning of human limbs

The first issue to take into account when defining a set of cascades of classifiers is how to define the groups of limbs to be learnt by each individual cascade. For this task, we propose to train a tree-structure cascade of classifiers. This tree-structure defines the set of meta-classes for each dichotomy (cascade of classifiers) taking into account the visual appearance of body parts, which has two purposes. On one hand, we aim to avoid dichotomies in which body parts with different visual appearance belong to the same meta-class. On the other hand, the dichotomies that deal with classes that are difficult to learn (body parts with similar visual appearance) are defined taking into account few classes. An example of the body part tree-structure defined taking into account these issues for a set of 7 body limbs is shown in Fig. 6.3(a). Notice that classes with similar visual appearance (e.g. upper-arm and lower-arm) are grouped in the same meta-class in most dichotomies. In addition, dichotomies that deal with difficult problems (e.g. $\mathbf{m}_5$) are focused only in the difficult classes, without taking into account all other body parts. In this case, class c^7 denotes the background.

6.3.3 ECOC multi-limb detection

In the ECOC framework, given a set of k classes (body parts) to be learnt, l different bi-partitions (groups of classes or dichotomies) are formed, and l binary problems over the partitions are trained [12]. As a result, a codeword of length l is obtained for each class, where each position (bit) of the code corresponds to a response of a given classifier h^i (coded by $+1$ or -1 according to their class set membership, or 0 if a particular class is not considered for a given classifier). Arranging the codewords as rows of a matrix, we define a *coding matrix* $\mathbf{M}$, where $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$. During the *decoding* (or testing) process, applying the l binary classifiers, a code $\mathbf{f}(\mathbf{x}_t)$ is obtained for each data sample $\mathbf{x}_t$ in the test set. This code is compared to the codewords $(\mathbf{m}^i, i \in [1, .., k])$ of each class defined in the matrix $\mathbf{M}$, and the data sample is assigned to the class with the *closest* codeword [50].

The ECOC coding step has been widely tackled in the literature either by predefined or problem-dependent strategies. However, recent works showed that problem-dependent strategies can obtain high performance by focusing on the idiosyncrasies of the problem [11]. Following this fashion, we define a problem dependent coding matrix in order to allow the inclusion of cascade of classifiers and learn the body parts. In particular, we propose to use a predefined *coding* matrix in which each dichotomy is obtained from the body part tree-structure described in previous section. Fig. 6.3(b) shows the coding matrix codification of the tree-structure in Fig. 6.3(a).

Loss-weighted decoding using cascade of classifier weights

In the ECOC *decoding* step an image is processed using a windowing method, and then, each image patch, that is, a sample $\mathbf{x}$, is described and tested. In this sense, each classifier f^i outputs a prediction whether $\mathbf{x}$ belongs to one of the two previously learnt meta-classes. Once the set of predictions $\mathbf{f}(\mathbf{x})$ is obtained, it is compared to the set of codewords of M, using a decoding function $\delta(\mathbf{M}, \mathbf{f}(\mathbf{x}))$. Thus, the final prediction is the class with the codeword that minimizes $\underset{i}{\operatorname{argmin}}\, \delta(\mathbf{m}^i, \mathbf{f}(\mathbf{x}))$. In [50] the authors proposed a problem-dependent decoding function (distance function that takes into account classifier performances) obtaining very satisfying results. Following this core idea, we use the Loss-Weighted decoding of Equation 6.1, where $\mathbf{W}$ is a matrix of weights and $\mathcal{L}$ is a loss function ($\mathcal{L}(\theta) = \exp^{-\theta}$).

$$\delta_{LW}(\mathbf{m}^i, \mathbf{f}(\mathbf{x})) = \sum_{j=1}^{l} w_{ij} \mathcal{L}(m_{ij} \cdot f^j(\mathbf{x}))) \tag{6.1}$$

In Equation 6.1, $\mathbf{W}$ (weight matrix) corresponds to the product of cascade accuracies at each stage. Thus, each column of $\mathbf{W}$ is assigned a weight $\mathbf{w}_i$ as,

$$\mathbf{w}_i = \prod_{j=1}^{s} \frac{TP(h_{ij}) + TN(h_{ij})}{TP(h_{ij}) + FN(h_{ij}) + FP(h_{ij}) + TN(h_{ij})}, \tag{6.2}$$

for a cascade of classifiers of s stages, where h_{ij} stands for the i-th cascade and stage j, $j \in [1, .., s]$, and TP, TN, FN, and FP computes the number of true positives, true negatives, false negatives and false positives, respectively.

Finally, a body-like confidence map $\mathbf{B} \in [0,1]^{u \times v}$, is computed for each image where u and v are the length and width of the image. This map contains, at each position b_{ij}, the proportion of bodypart detections for each pixel over the total number of detections for the whole image. In other words, pixels belonging to the human body will show a higher body-like confidence than the pixels belonging to the background.

6.3.4 Binary GraphCut optimization for foreground mask extraction

GraphCuts [68, 114] has been widely used for image segmentation in various problems [22, 24, 25]. Intuitively, GraphCuts represents an image as a graph and defines a trimap using prior probability class distributions over the image. To find a segmentation, GraphCut minimizes an energy function which is composed by a unary potential term (often taken as a prior knowledge over image pixels) and a neighborhood potential term. With this energy minimization scheme and given the initial map, the final segmentation is performed using a minimum cut algorithm. However, we propose to omit the classical semiautomatic initialization by an automatic assignment based on the human body confidence map $\mathbf{B} \in [0,1]^{l \times w}$. In this sense, depending on the probability of each pixel it will be assigned to a certain class, either *background* or *body*. For a more detailed explanation and formulation (which falls far out of the scope of this thesis) we refer readers unexperienced on GraphCuts to the seminal paper of Boykov et. al [23].

6.3.5 Tree-structure body part learning without background

Once the binary person/background segmentation is performed by means of GraphCut (mask shown in Fig. 6.2(e)), we apply a second procedure in order to split the person mask into a set of human limbs. For this step, we define a new tree-structure classifier equivalent to the one described in Section 6.3.2 without including the background class c^7 shown in Fig. 6.3(a).

6.3.6 ECOC multi-limb detection

In order to obtain an accurate detection of human limbs we take profit of the HOG descriptor [35] and SVM classifier which have shown to obtain robust results in human estimation scenarios [35, 57, 68]. We extract HOG features for the different body parts (previously normalized to dominant region orientation), and then, SVMs classifiers are trained on that feature space, using a Generalized Gaussian RBF Kernel based on Chi-squared distance [137]. This stage follows a similar pipeline as the one described in Section 6.3.3. In this sense, each SVM classifier learns a binary partition of human limbs but without taking into account the background class. As shown in Fig. 6.3(b), we train $l = 6$ SVMs with different binary human-limb partitions.

At the ECOC decoding step, we use the Loss-Weighted decoding [50] function shown in Equation 4.9 (an example is shown in Fig 6.3(b)). In this sense, for each RGB test image corresponding to the binary mask shown in Fig. 6.2(e), we adopt a sliding window approach and test each patch on our ECOC multi limb recognition system. Then, based on the ECOC output we construct a set of limb-like probability maps. Each map $\mathbf{B}^c$ contains, at each position b_{ij}^c, the confidence of pixel at the entry (i,j) of belonging to the body part class c, where $c \in \{1, 2, ..., k\}$. This probability is computed as the proportion of positive detections at point (i,j) over all detection for class c. Examples of probability maps obtained from ECOC outputs are shown in Fig. 6.2(h). While Haar-like based on AdaBoost gave us a very accurate and fast initialization of human regions for binary user segmentation, in this second step, HOG-SVM is applied in a reduced region of the image, providing better estimates of human limb locations.

6.3.7 Alpha-beta swap Graph Cuts multi-limb segmentation

Once the likelihood maps for each limb have been obtained the final step is to compute the multi-limb segmentation. We base our proposal on Graph Cuts theory to tackle our human-limb segmentation problem [22, 23, 25, 68, 114]. In [25], Boykov et. al. developed an algorithm, named α-β swap graph-cut, which is able to cope with the multi-label segmentation problem. The α-β swap graph-cut is an extension of binary graph cuts that performs an iterative procedure where each pair of labels (c_q, c_m), $\{m, q\} \in \{1, 2, ..., k\}$, are segmented using GraphCuts. This procedure segment all α pixels from β pixels with GraphCuts and the algorithm will update the α-β combination at each iteration until convergence.

In this sense, an initial labeling is defined by an automatic trimap assignment based on the set of limb-like probability maps $\mathbf{B}^c \in [0, 1]^{l \times w}$ defined in previous section. In addition, to penalize relations between pixels z_q and z_m depending on their label assignations, a user-predefined pair-wise cost to each possible combination of labels $\Omega(c_q, c_m)$ is introduced.

In concrete, in order to introduce prior costs between different labels, $\Omega(c_q, c_m)$ must fulfill some constraints related to spatial coherence between the different labels, taking into account the natural constraints of the human limbs (i.e. head must be closer to torso than legs, arms are nearer to forearms than head, etc.). In particular, we experimentally fixed the penalization function Ω as defined in Table 6.2.

	Head	**Torso**	**Arms**	**Forearms**	**Thighs**	**Legs**	**Background**
Head	0	20	35	50	70	90	1
Torso	20	0	15	25	40	70	1
Arms	35	15	0	10	60	80	1
Forearms	50	25	10	0	30	60	1
Thighs	70	40	60	30	0	10	1
Legs	90	70	80	60	10	0	1
Background	1	1	1	1	1	1	1

Table 6.2: Prior cost between each pair of labels.

6.4 Experimental results

In order to present the experimental results, we first discuss the data, experimental settings, methods and validation protocol.

6.4.1 Data

We use the proposed HuPBA 8k+ dataset described in Section 6.2. We reduced the number of limbs from the 14 available in the dataset to 6, grouping those that are similar by symmetry (right-left) as arms, forearms, thighs and legs. Thus, the set of limbs of our problem is composed by: *head*, *torso*, *forearms*, *arms*, *thighs* and *legs*. Although labeled within the dataset, we did not include hands and feet in our multi-limb segmentation scheme. Finally, in order to train the limb classifiers, ground truth masks are used to normalize all limb regions per dominant orientation, and both Haar-like features and HOG descriptors are computed based on the aspect ratio of each region, making the descriptions scale invariant.

6.4.2 Methods and experimental settings

To evaluate the performance of our proposal for multi-limb segmentation, we compare our strategy with two state-of-the-art methods for multi-limb segmentation:

- **FMP:** This method was proposed by Yang and Ramanan [138, 139] and it is based on Flexible Mixtures-of-Parts (FMP). We compute the average of each set of mixtures for each limb and for each pyramid level in order to obtain the probability maps for each limb category. In order to compute the probability map of the background category, we subtract 1 with the maximum probability $\in [0, 1]$ of the set of limbs detection at pixel location.
- **IPP:** This method is proposed by Ramanan [108] and it is based on an Iterative Parsing Process (IPP). We use it to extract the limb-like probability maps followed by α-β swap graph-cut multi-limb segmentation. The background category is computed as described for the FMP method.
- **ECOC+GraphCut:** Our proposed human limb segmentation scheme shown in Fig. 6.2.

Experimental settings

In a preprocessing step, we resized all limb samples to a 32×32 pixels region for computational purposes. Then, we used the standard Cascade of Classifiers based on AdaBoost and Haar-like features [131], and we forced a 0.99 false positive rate and maximum of 0.4 false alarm rate during 8 stages. To detect limbs with trained cascades of classifiers, we applied a sliding window approach with an initial patch size of 32×32 pixels up to 60×60 pixels. As a final part of the first stage, binary Graph Cuts were applied to obtain the binary segmentation where the initialization values of foreground and background were provided to the binary Graph Cut algorithm and tuned via cross-validation.

For the second stage, we set the following parameters for the HOG descriptor: 32×32 window size, 16×16 block size, 8×8 block stride, 8×8 cell size and 8 for number of bins. Then, we trained SVMs with a Generalized Gaussian RBF kernel based on Chi-squared distance, (see Fig.(a) 6.3). The parameters of the kernel, C and γ were tuned via cross-validation. Finally, the model selection step was done via a leave-one-sequence-out cross-validation. For multi-limb segmentation we used the alpha-beta GraphCut procedure, where we set a 8×8 neighboring grid and tuned the λ parameter of GraphCut using cross-validation.

6.4.3 Validation measurement

In order to evaluate the results for the two different tasks: binary segmentation and multi-label segmentation, we use the Jaccard Index of overlapping ($J = \frac{A \bigcap B}{A \bigcup B}$) where A is the ground-truth and B is the corresponding prediction.

6.4.4 Multi-limb segmentation results

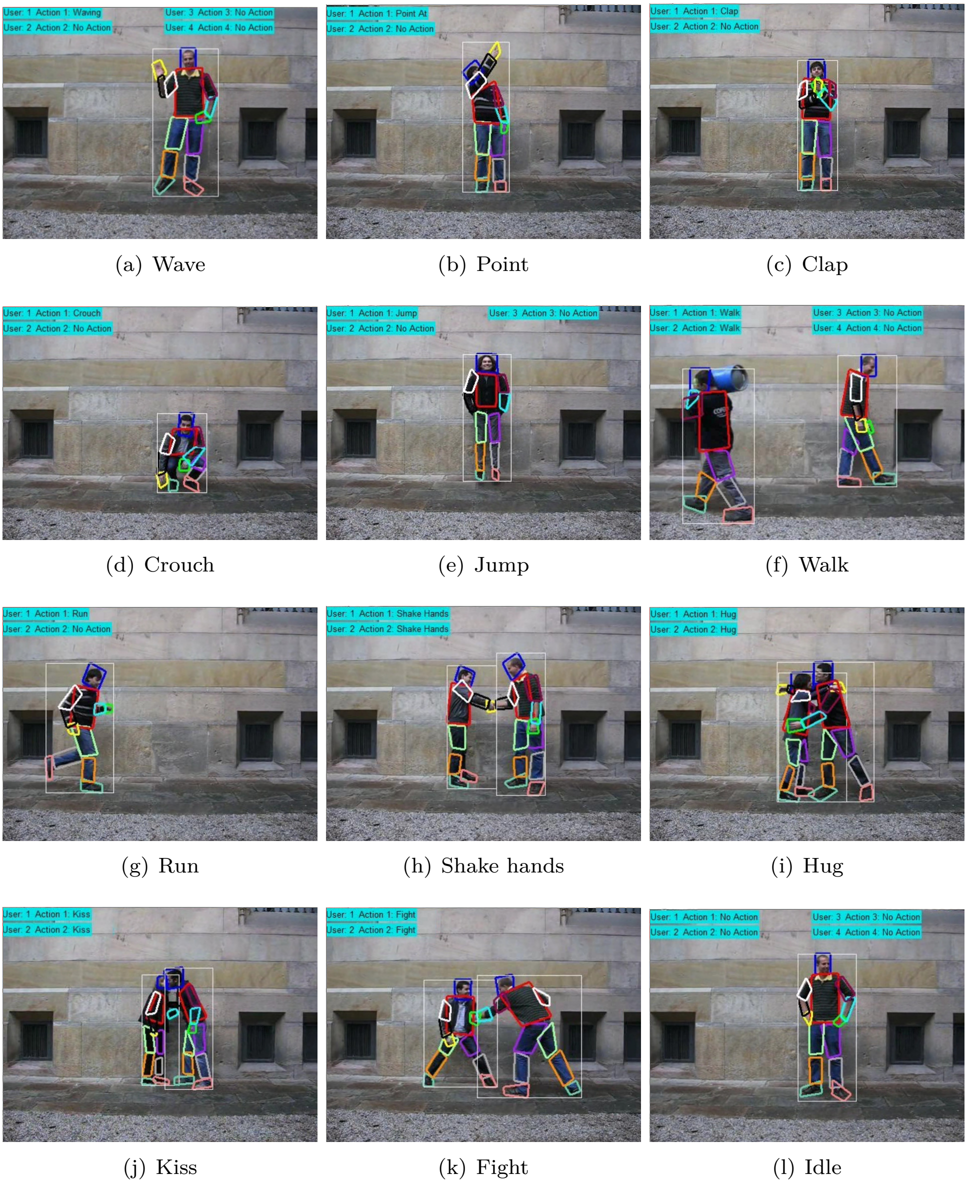
For the Multi-limb segmentation task, we show in Fig. 6.4 and Fig. 6.5 qualitative results for some samples of the *HuPBA* 8*k*+ dataset. When comparing the qualitative results we can see how the FMP method [138, 139] performs worse than its counter parts. In addition, one can se how IPP and our method obtain similar results in most cases. However, the IPP lacks of a good person/background segmentation.

Furthermore, we provide with quantitative results in terms of the Jaccard Index. In Fig. 6.6 we show the overlapping performance obtained by the different methods, where each plot shows the overlapping for a certain limb. In addition, we use a ’Do not care’ value which provides a more flexible interpretation of the results. Consider the ground truth of a certain limb category in an image as a binary image, which pixels take value 1 when those pixels are labeled to belong to such limb. Then, the ’Do not care’ value is defined as the number of pixels which are ignored at the limits of each one of the ground truth instances. Thus, by using this approach we can compensate the pessimistic overlap metric in situations when the detection is shifted some pixels. In this sense, we analyze the overlapping performance as a function of a ’Do not care’ value that ranges from 0 to 4.

When analyzing quantitative results, we see how our method outperforms the compared methodologies for some limb categories. In particular, for the *head* region both methods obtain similar results, which is intuitive since the method used to detect the head is the well known face detector. Finally, we see how FMP method is in almost all cases obtaining the worst performance. As shown in Figure 12(g), for the mean overlapping considering all the segmented limbs our method outperforms the rest of approaches up to 3 pixels of "Do not care" evaluation.

6.5 Conclusions

In this chapter of the thesis we proposed a novel ECOC-based two-stage method for human multi-limb segmentation in RGB images. In the first stage, we perform a person/background segmentation by training a set of body parts using cascades of classifiers embedded in an ECOC framework. In the second stage, to obtain a multi-limb segmentation we applied multi-label GraphCuts to a set of limb-like probability maps obtained from a problem-dependent ECOC scheme. Furthermore, we introduced the *HuPBA* $8K+$ dataset, which represents the largest available multi-limb dataset on RGB data up to date, with more than 120000 manually labeled limb regions. We compared our ECOC-based proposal with state-of-the-art pose-recovery approaches on the novel dataset, obtaining very satisfying results.


User: 1 Action 1: Waving
User: 2 Action 2: No Action
User: 3 Action 3: No Action
User: 4 Action 4: No Action
User: 1 Action 1: Point At
User: 2 Action 2: No Action
User: 1 Action 1: Clap
User: 2 Action 2: No Action
User: 1 Action 1: Crouch
User: 2 Action 2: No Action
User: 1 Action 1: Jump
User: 2 Action 2: No Action
User: 3 Action 3: No Action
User: 1 Action 1: Walk
User: 2 Action 2: Walk
User: 3 Action 3: No Action
User: 4 Action 4: No Action
User: 1 Action 1: Run
User: 2 Action 2: No Action
User: 1 Action 1: Shake Hands
User: 2 Action 2: Shake Hands
User: 1 Action 1: Hug
User: 2 Action 2: Hug
User: 1 Action 1: Kiss
User: 2 Action 2: Kiss
User: 1 Action 1: Fight
User: 2 Action 2: Fight
User: 1 Action 1: No Action
User: 2 Action 2: No Action
User: 3 Action 3: No Action
User: 4 Action 4: No Action

(a) Wave (b) Point (c) Clap

(d) Crouch (e) Jump (f) Walk

(g) Run (h) Shake hands (i) Hug

(j) Kiss (k) Fight (l) Idle

Figure 6.1: Different samples labeled on the HuPBA $8k+$ dataset.

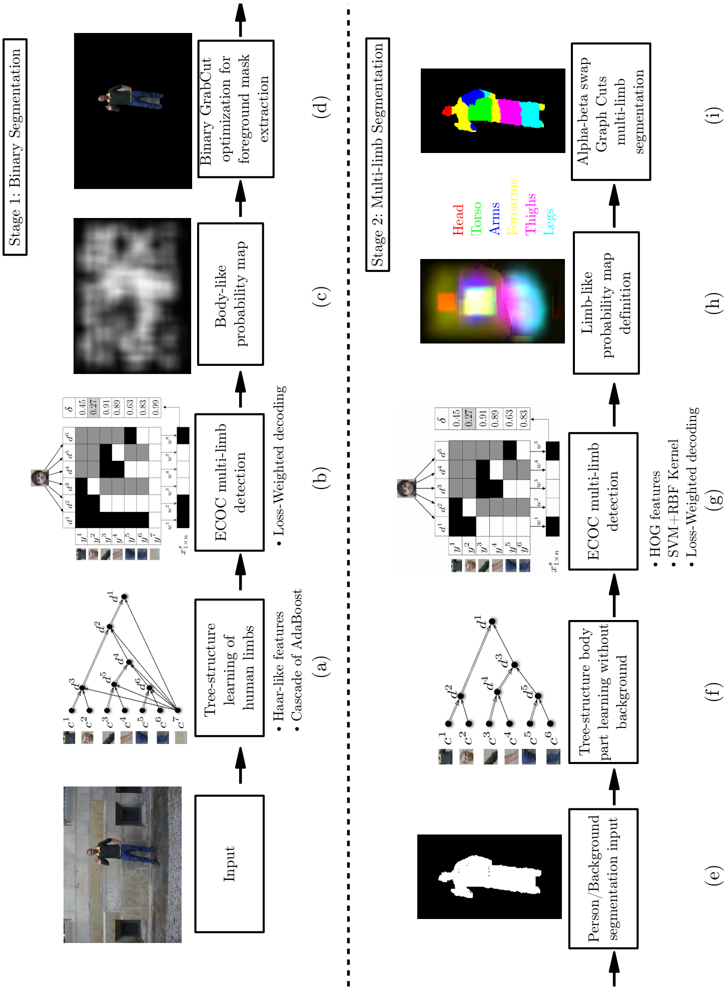
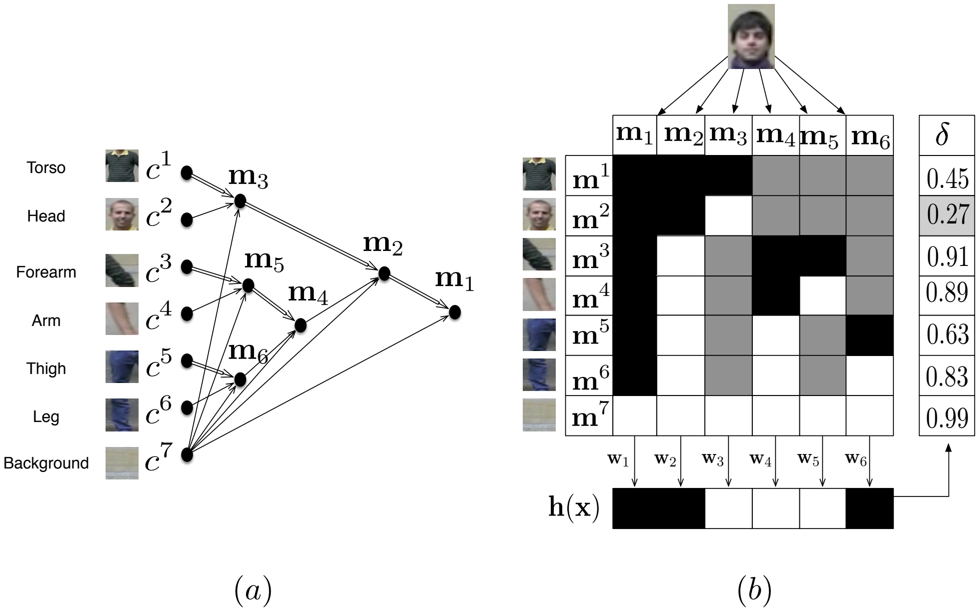

Stage 1: Binary Segmentation
Input
Tree-structure learning of human limbs
• Haar-like features
• Cascade of AdaBoost
(a)
ECOC multi-limb detection
• Loss-Weighted decoding
(b)
δ
0.45
0.27
0.91
0.89
0.63
0.83
0.99
Body-like probability map
(c)
Binary GrabCut optimization for foreground mask extraction
(d)
Stage 2: Multi-limb Segmentation
Person/Background segmentation input
(e)
Tree-structure body part learning without background
(f)
ECOC multi-limb detection
• HOG features
• SVM+RBF Kernel
• Loss-Weighted decoding
(g)
δ
0.45
0.27
0.91
0.89
0.63
0.83
Limb-like probability map definition
(h)
Head
Torso
Arms
Forearms
Thighs
Legs
Alpha-beta swap Graph Cuts multi-limb segmentation
(i)

Figure 6.2: Scheme of the proposed human-limb segmentation method.



	$\mathbf{m}_1$	$\mathbf{m}_2$	$\mathbf{m}_3$	$\mathbf{m}_4$	$\mathbf{m}_5$	$\mathbf{m}_6$	δ
$\mathbf{m}^1$	■	■	■	▨	▨	▨	0.45
$\mathbf{m}^2$	■	■	□	▨	▨	▨	0.27
$\mathbf{m}^3$	■	□	▨	■	■	▨	0.91
$\mathbf{m}^4$	■	□	▨	■	□	▨	0.89
$\mathbf{m}^5$	■	□	▨	□	▨	■	0.63
$\mathbf{m}^6$	■	□	▨	□	▨	□	0.83
$\mathbf{m}^7$	□	□	□	□	□	□	0.99
$\mathbf{h}(\mathbf{x})$	■	■	□	□	□	■	

Figure 6.3: (a) Tree-structure classifier of body parts, where nodes represent the defined dichotomies. Notice that the single or double lines indicate the meta-class defined. (b) ECOC decoding step, in which a head sample is classified. The coding matrix codifies the tree-structure of (a), where black and white positions are codified as +1 and −1, respectively.

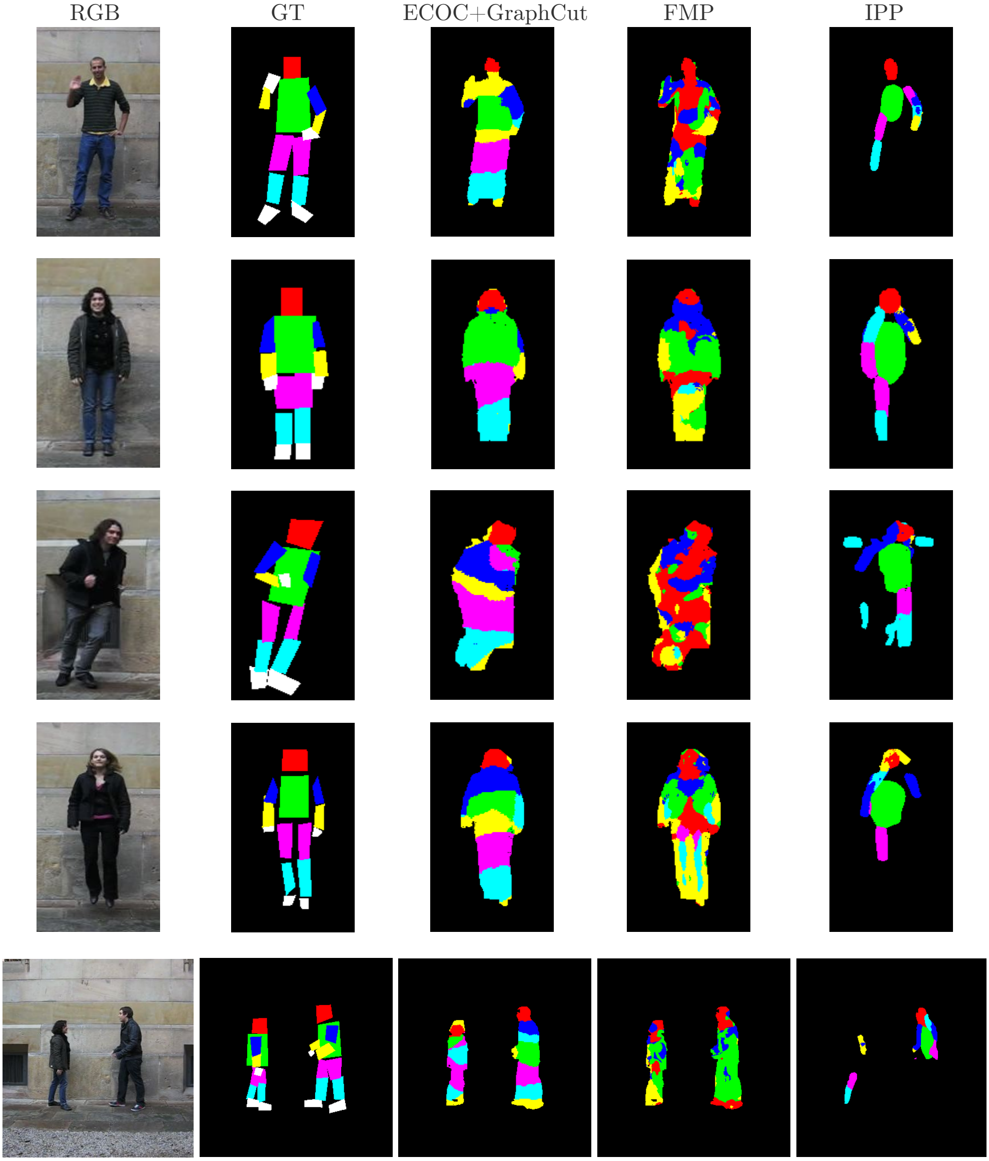


Figure 6.4: Multi-limb segmentation results for the three methods, for each sample, we also show the RGB image and the ground-truth (GT).

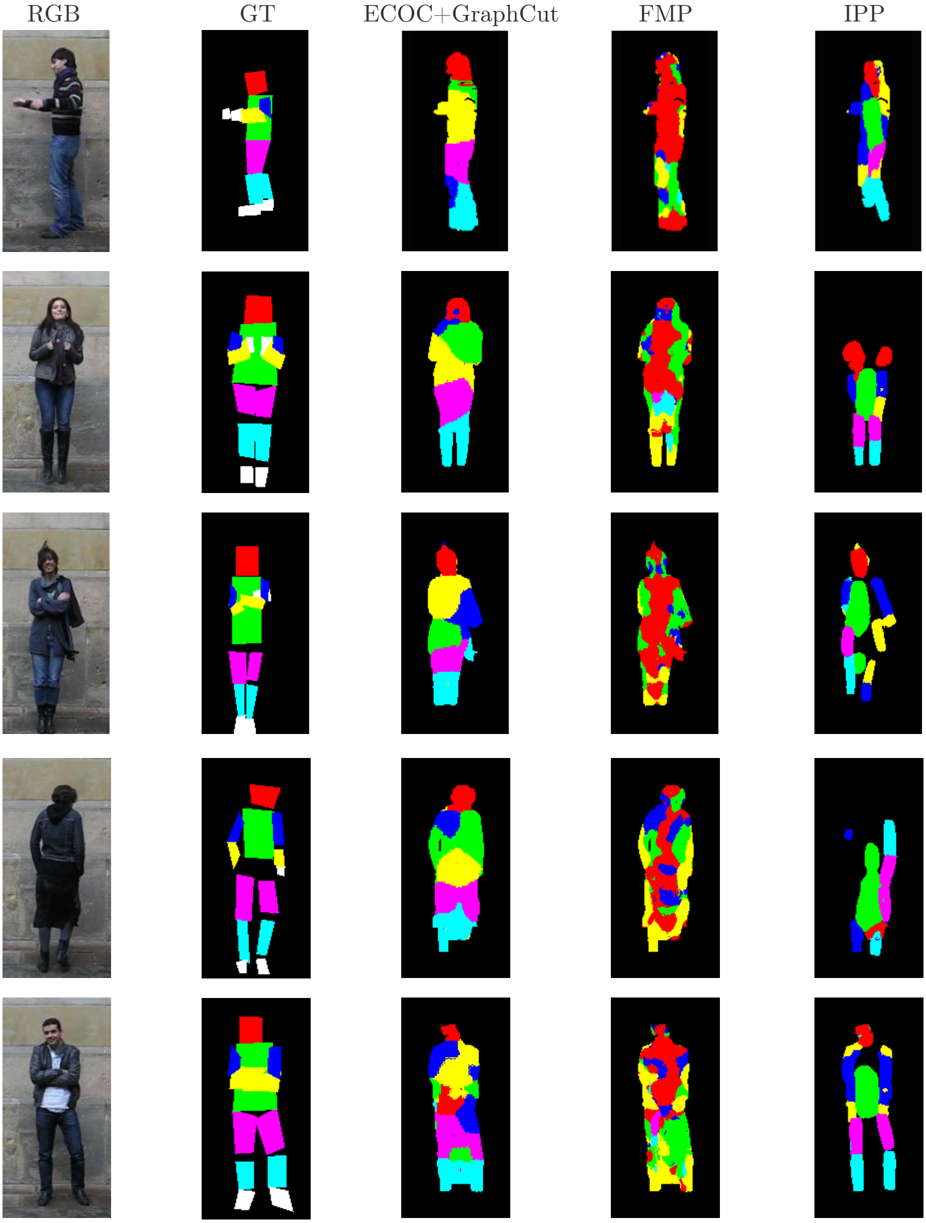
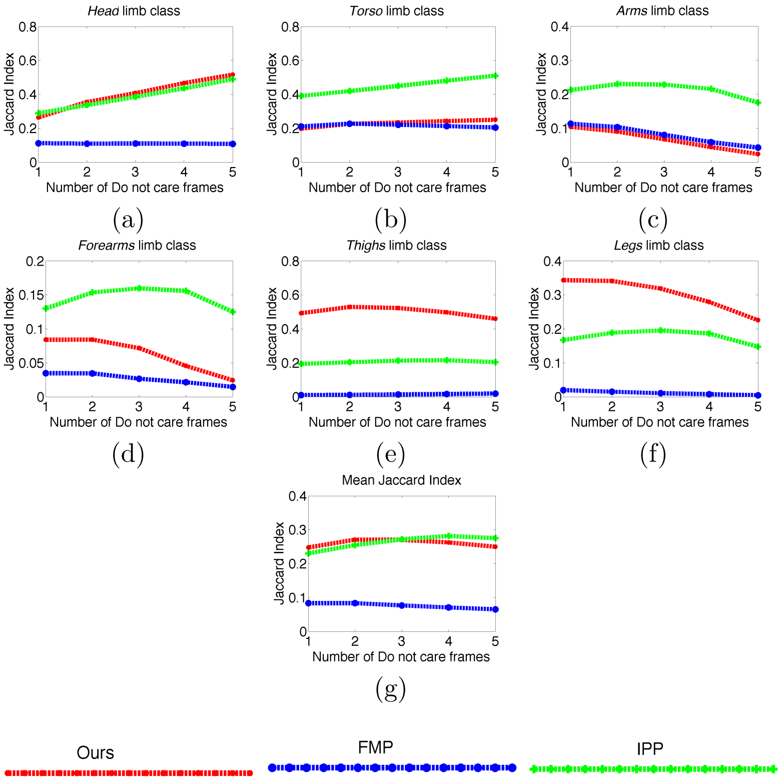


Figure 6.5: Multi-limb segmentation results for the three methods, for each sample, we also show the RGB image and the ground-truth (GT).



(a) (b) (c)

(d) (e) (f)

(g)

Figure 6.6: Jaccard Indexes for the different limb categories from (a) to (f). (g) Mean Jaccard Index among all limb categories.

Part IV

Epilogue

Chapter 7

Conclusions

Multi-class problems arise very often in real-life situations. In this dissertation we have presented several ways to treat this multi-class problems by means of Error-Correcting Output Codes. In particular, we have focused in the critical aspect of ECOC coding designs, contributing with different problem-dependent coding designs that take into account the structure of multi-class data, in order to optimize the ECOC coding design to the multi-class data we proposed two different schemes for genetic optimization. In addition, as shown in earlier chapters of this dissertation we also have presented a new way to represent ECOC coding designs, which enables to study the error-correcting capabilities between pairs of classes. Enabled by this novel representation we introduce a discrete matrix factorization algorithm to learn ECOC coding matrices.

7.1 Summary of contributions

The coding step of the ECOC framework is critical for its performance and generalization capabilities. In this sense, our contributions in this dissertation are the following:

1. **ECOC separability and error-correction**: The error-correcting capability of an ECOC has always been depicted in literature as a single scalar, which hinders further analyses of the distribution of error-correction between different categories. In this dissertation we proposed to represent an ECOC by means of its separability matrix and use very simple heuristics to exploit the distribution of error-correction among pairs of classes to outperform state-of-the-art results.
2. **Minimal ECOC Coding Design**: We proposed to define the lower bound of an ECOC design in terms of the number of binary problems embedded in the coding design. We showed that even with a very reduced number of binary classifiers we can obtain comparable results to state-of-the-art coding designs.
3. **Genetic Optimization of Minimal ECOCs**: Although showing a high performance, Minimal ECOCs are bounded in terms of generalization capabilities due to the reduced number of classifiers. In this sense, we proposed to use a Genetic Algorithm to optimize the Minimal ECOC coding configuration and obtain a Minimal ECOC coding matrix with high generalization capabilities.
4. **ECOC-Compliant Genetic Algorithm**: Standard Genetic Algorithm use crossover and mutation operators that treat individuals as binary strings. This operators do not

take into account the constraints of the ECOC framework and can lead to poor optimization schemes. In this dissertation we proposed to redefine the standard crossover and mutation operators in order to take into account the constraints of the ECOC framework. In addition we also proposed and operator to dynamically adapt the code length of the ECOC during the training process. This redefinition of operators led to very satisfying results in terms of classification performance as a function of the model complexity.

5. **Error-Correcting Factorization**: Empowered by the novel representation of an ECOC as its Separability Matrix we proposed to obtain an estimated Separability matrix from data using very simple statistics and use the proposed Error-Correcting Factorization to factorize the estimated Separability matrix into a discrete ECOC coding matrix that distributes error-correcting to those classes that are more prone to errors.
6. **Applications in Human Pose Recovery**: As a particular application, in this dissertation we applied the ECOC framework in the challenging problem of Human Pose Recovery and obtain very satisfying results in comparison with state-of-the-art works. In addition we proposed the HuPBA $8k+$ dataset, the biggest dataset for Human Pose Recovery in still images.

To conclude, in this dissertation we have defined novel problem dependent-design for the ECOC coding step that take into account the structure of the multi-class data in order to obtain models with high generalization performance with a reduced computational cost. In addition, we also presented a novel way to represent an analyze ECOC matrices that exploits the pair-wise error-correcting capabilities of an ECOC. The techniques obtain high generalization performance with a small code length, solving several multi-class real world problems like human pose recovery, face recognition or traffic sign categorization.

7.2 Future work

As lines of future research with potential impact we highlight the idea of error-correcting allocation. From a theoretical perspective this dissertation opens a wide set of questions about the distribution of error-correction of an ECOC design. While earlier state-of-the-art works proposed to allocate the error-correcting capabilities to those categories which are easily separable, we showed how by allocating error-correction to those classes prone to be confused we outperform state-of-the-art results. However, we consider that a best of both worlds solution can be interesting to pursue.

In regards to the proposed Error-Correcting Factorization one could potentially regularize the factorization problem in order to obtain sparse results or additional regularizations (i.e. nuclear norms). In addition, it would be very interesting to couple the ECF problem with the problem of learning the binary classifiers in order to obtain a joint optimization problem that obtains high generalization performance.

Moreover, the problem posed by ECF (matrix factorization in binary terms) does not only affect ECOC coding designs, but could impact other areas like Non-negative Matrix Factorization, Topic Learning, etc. In particular, the proposed ECF approach could be embedded in other classifiers that use codewords as their class outputs, for example, Convolutional Neural Networks.

For the particular application of Pose Recovery, the contextual relationship between parts play a very important role in prediction. In this sense, we consider that encoding contextual relationship between body parts by means of stacking outputs of several ECOC classifiers could dramatically boost performance.

Appendix A

Datasets

The data used for the experiments of different methods in this thesis consists of twelve multi-class datasets from the UCI Machine Learning Repository database [8]. The number of training samples, features, and classes per dataset are shown in Table A.1.

Table A.1: UCI repository datasets characteristics.

Problem	#Training samples	#Features	#Classes
Dermathology	366	34	6
Iris	150	4	3
Ecoli	336	8	8
Vehicle	846	18	4
Wine	178	13	3
Segmentation	2310	19	7
Glass	214	9	7
Thyroid	215	5	3
Vowel	990	10	11
Balance	625	4	3
Shuttle	14500	9	7
Satimage	4435	36	6
Yeast	1484	8	10

In this thesis, we also report results on five challenging Computer Vision problems: face recognition in the wild, traffic sign recognition, face recognition in controlled environments, music scores classification, and MPEG objects categorization. All the datasets are public and available upon request to the author.

For the face recognition in the wild we use *Labeled Faces in the Wild* [74] dataset to perform the multi-class face classification of a large problem consisting of 184 face categories. In addition, for the traffic sign recognition problem, we use the video sequences obtained from a Mobile Mapping System [27] to test the methods in a real *traffic sign categorization* problem consisting of 36 traffic sign classes. We also perform face recognition in controlled environments by using 20 classes from the *ARFaces* [95] dataset. We also apply the proposed methods in challenge handwritten music score classification classifying seven symbols from old scanned music scores, using the cleafs and accidental dataset [56]. Finally we classify the 70 visual object categories from the public *MPEG7 dataset* [1]. The characteristics of each dataset as described as follows.

- **Labeled Faces in the Wild categorization:** This dataset contains 13000 faces images taken directly from the web from over 1400 people. These images are not constrained in terms of pose, light, occlusions or any other relevant factor. For the purpose of this experiment we used a specific subset, taking only the categories which at least have four or more examples, having a total of 184 face categories. Finally, in order to extract relevant features from the images, we apply an Incremental Principal Component Analysis procedure [75], keeping 99.8% of the information. An example of face images is shown in Figure A.1.

Figure A.1: Labeled Faces in the Wild dataset.

- **Traffic sign categorization:** For this second computer vision experiment, we use the video sequences obtained from the Mobile Mapping System of [27] to test the Evolutionary Minimal ECOC methodology on a real traffic sign categorization problem. In this system, the position and orientation of the different traffic signs are measured with video cameras fixed on a moving vehicle. 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. From this system, a set of 36 circular and triangular traffic sign classes are obtained. Some categories from this data set are shown in Figure A.2. The dataset contains a total of 3481 samples of size 32×32, filtered using the Weickert anisotropic filter, masked to exclude the background pixels, and equalized to prevent the effects of illumination changes. These feature vectors are then projected into a 100 feature vector by means of PCA.

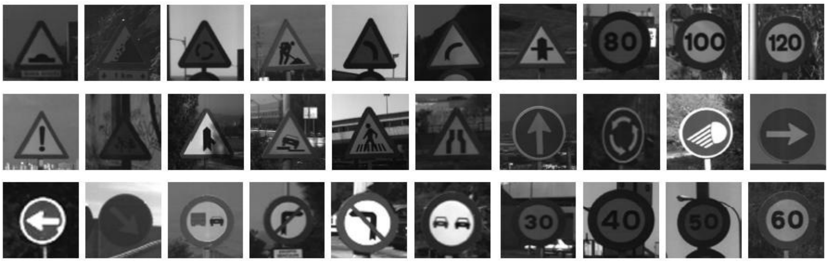


Figure A.2: Traffic sign classes.

- **ARFaces classification:** The ARFace database [95] is composed of 26 face images from 126 different subjects (70 men and 56 women). The images have uniform white background. The database has two sets of images from each person, acquired in two different sessions, with the following structure: one sample of neutral frontal images, three samples with strong changes in the illumination, two samples with occlusions (scarf and glasses), four images combining occlusions and illumination changes, and three samples with gesture effects. One example of each type is plotted in Figure A.3. For this experiment, we selected all the samples from 20 different categories (persons).

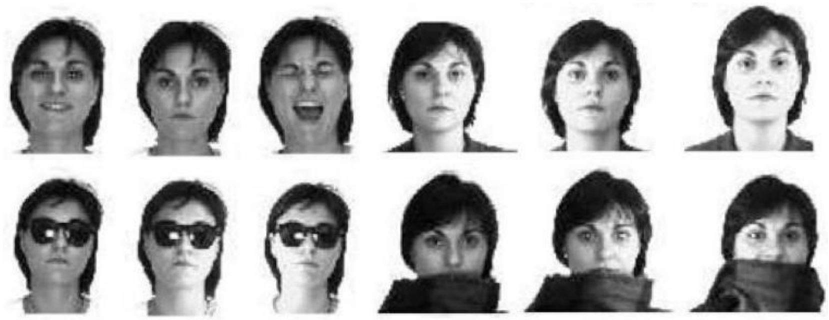
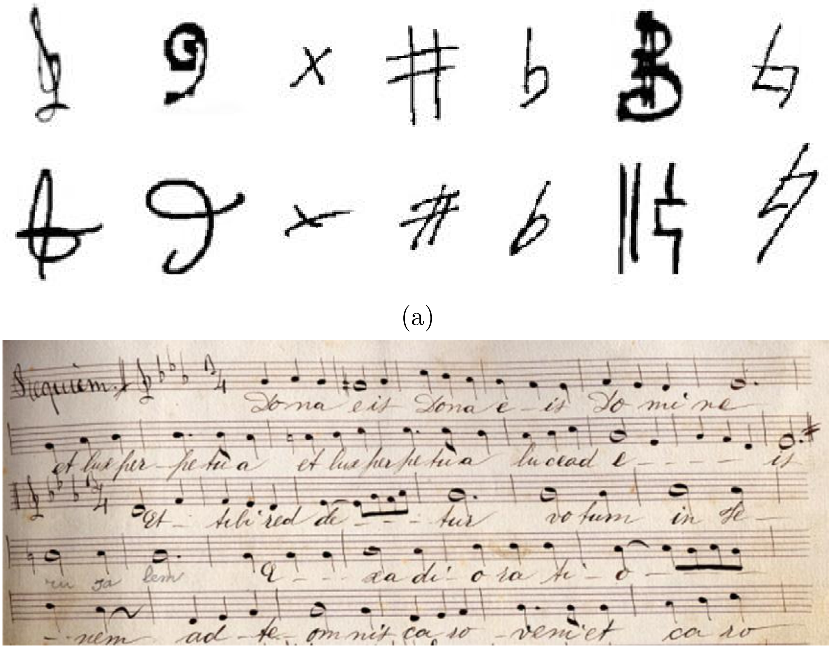


Figure A.3: ARFaces dataset classes. Examples from a category with neutral, smile, anger, scream expressions, wearing sun glasses, wearing sunglasses and left light on, wearing sun glasses and right light on, wearing scarf, wearing scarf and left light on, and wearing scarf and right light on.

- **Clefs and accidental dataset categorization:**

  The dataset of clefs and accidental [56] is obtained from a collection of modern and old musical scores (19th century) of the Archive of the Seminar of Barcelona. The dataset contains a total of 4098 samples among seven different types of clefs and accidental from 24 different authors. The images have been obtained from original image documents using a semi-supervised segmentation approach [56]. The main difficulty of this dataset is the lack of a clear class separability because of the variation of writer styles and the absence of a standard notation. A pair of segmented samples for each of the seven classes showing the high variability of clefs and accidental appearance from different authors can be observed in Figure A.4 (a). An example of an old musical score used to obtain the data samples are shown in Figure A.4(b). The object images are described using the Blurred Shape Model descriptor [45].
- **MPEG7 categorization:** The MPEG7 dataset contains 70 classes with 20 instances per class, which represents a total of 1400 object images. All samples are described using the Blurred Shape Model descriptor [45]. A couple of samples for some categories of this dataset are shown in Figure A.5.



Requiem
Dona eis Dona e - is Do mi ne
et lux per - pe tu a et lux per pe tu a luceat e - - - - is
Et - tibi red de - - - tur votum in Je -
ru sa lem e - - xa di o ra ti - o -
- nem ad te omnis ca ro veniet ca ro

(a)

(b)

Figure A.4: (a) Object samples. (b) Old music score.

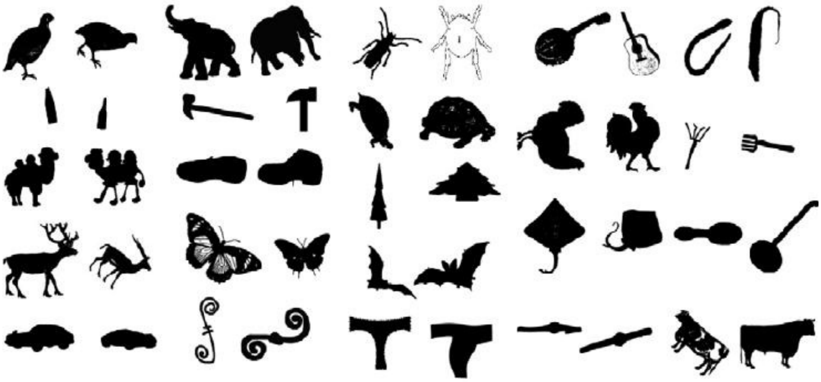


Figure A.5: MPEG7 samples.

Appendix B

Publications

The following publications are a consequence of the research carried out during the elaboration of this thesis and give an idea of the progression that has been achieved.

B.1 Journals

- **Error-Correcting Factorization**, Bautista, Miguel Ángel and De la Torre, Fernando and Pujol, Oriol, and Escalera, Sergio. IEEE Transactions on Pattern Analysis and Machine Intelligence. Under Review, 2015.
- **A Gesture Recognition System for Detecting Behavioral Patterns of ADHD**, Bautista, Miguel Ángel and Hernández-Vela, Antonio and Escalera, Sergio and Igual, Laura and Pujol, Oriol and Moya, Josep and Violant, Verónica and Anguera, María Teresa, IEEE Transactions on System, Man and Cybernetics, Part B. In Press, 2015.
- **On the design of an ECOC-compliant genetic algorithm**, Bautista, Miguel Ángel and Escalera, Sergio and Baró, Xavier and Pujol, Oriol, Pattern Recognition, 47, 2, 865–884, 2013, Elsevier.
- **HuPBA 8k+: Dataset and ECOC-GraphCut based Segmentation of Human Limbs**, Sanchez, Daniel and Bautista, Miguel Ángel and Escalera, and Sergio. Neurocomputing, 2014, Elsevier.
- **Minimal Design of Error-Correcting Output Codes**, Bautista, Miguel Ángel and Escalera, Sergio and Baró, Xavier and Radeva, Petia and Vitriá, Jordi and Pujol, Oriol. Pattern Recognition Letters, 33, 6, 693–702, 2011, Elsevier.
- **Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d**, Hernández-Vela, Antonio and Bautista, Miguel Ángel and Perez-Sala, Xavier and Ponce-López, Víctor and Escalera, Sergio and Baró, Xavier and Pujol, Oriol and Angulo, Cecilio. Pattern Recognition Letters, 50, 112-121, 2014, Elsevier.

B.2 International Conferences and Workshops

- **Introducing the separability matrix for error correcting output codes coding**. Bautista, Miguel and Pujol, Oriol and Baró, Xavier and Escalera, Sergio. Multiple Classifier Systems, 227–236, 2011, Springer.

- **On the Design of Low Redundancy Error-Correcting Output Codes**. Bautista, Miguel and Escalera, Sergio and Baró, Xavier and Pujol, Oriol and Vitria, Jordi and Radeva, Petia. Ensembles in Machine Learning Applications, 21–38, 2011, Springer.
- **Compact evolutive design of error-correcting output codes**. Bautista, MA and Baro, X and Pujol, O and Radeva, P and Vitria, J and Escalera, S. European Conference on Machine Learning Workshops, 119–128, 2010, LNCS.
- **Probability-based Dynamic Time Warping for Gesture Recognition on RGB-D data**. Bautista, Miguel Angel and Hernández-Vela, Antonio and Ponce, Victor and Perez-Sala, Xavier and Baró, Xavier and Pujol, Oriol and Angulo, Cecilio and Escalera, Sergio. International Conference on Pattern Recogntion Workshops, Advances in Depth Image Analysis and Applications, 126–135, 2012, LNCS 7854 - Springer.
- **A genetic inspired optimization for ECOC**. Bautista, Miguel Ángel and Escalera, Sergio and Baró, Xavier and Pujol, Oriol. Structural, Syntactic, and Statistical Pattern Recognition, 743–751, 2012, Springer Berlin Heidelberg.
- **BoVDW: Bag-of-Visual-and-Depth-Words for gesture recognition**. Hernández-Vela, Antonio and Bautista, Miguel Angel and Perez-Sala, Xavier and Ponce, Victor and Baró, Xavier and Pujol, Oriol and Angulo, Cecilio and Escalera, Sergio. 21st International Conference on Pattern Recognition (ICPR), 2012, 449–452, 2012, IEEE.
- **Human body segmentation with multi-limb error-correcting output codes detection and graph cuts optimization**. Sánchez, Daniel and Ortega, Juan Carlos and Bautista, Miguel Ángel and Escalera, Sergio. Pattern Recognition and Image Analysis, 50–58, 2013, Springer Berlin Heidelberg.
- **Chalearn looking at people challenge 2014: Dataset and results**. Escalera, Sergio and Bar, X and Gonzlez, J and Bautista, Miguel A and Madadi, Meysam and Reyes, Miguel and Ponce, V and Escalante, Hugo J and Shotton, Jamie and Guyon, Isabelle. IEEE European Conference Computer Vision Looking At People Workshop, 2014.
- **Learning To Segment Humans By Stacking Their Body Parts**. Puertas, Eloi and Bautista, MA and Sanchez, Daniel and Escalera, Sergio and Pujol, Oriol. IEEE European Conference Computer Vision Looking At People Workshop, 2014.

Bibliography

[1] http://www.cis.temple.edu/latecki/research.html.

[2] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J. Kriegman, and S. Belongie. Generalized non-metric multidimensional scaling. In *International Conference on Artificial Intelligence and Statistics*, pages 11–18, 2007.

[3] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *JMLR*, 1:113–141, 2002.

[4] E. L. Allwein, R. E. Schapire, Y. Singer, and P. Kaelbling. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[5] E. Alpaydin and E. Mayoraz. Learning error-correcting output codes from data. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 2, pages 743 –748 vol.2, 1999.

[6] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009.

[7] M. Ángel Bautista, S. Escalera, X. Baró, and O. Pujol. On the design of an ecoc-compliant genetic algorithm. *Pattern Recognition*, 47(2):865 – 884, 2013.

[8] A. Asuncion and D. 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. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In A. Prieditis and S. Russel, editors, *The Int. Conf. on Machine Learning 1995*, pages 38–46, San Mateo, CA, 1995. Morgan Kaufmann Publishers.

[10] M. Bautista, S. Escalera, X. Baró, P. Radeva, J. Vitriá, and O. Pujol. Minimal design of error-correcting output codes. *Pattern Recognition Letters. In press.*, 2011.

[11] M. Á. Bautista, S. Escalera, X. Baró, and O. Pujol. On the design of an ecoc-compliant genetic algorithm. *Pattern Recognition*, 47(2):865–884, 2014.

[12] M. A. Bautista, S. Escalera, X. Baró, P. Radeva, J. Vitriá, and O. Pujol. Minimal design of error-correcting output codes. *Pattern Recogn. Lett.*, 33(6):693–702, Apr. 2012.

[13] U. R. Beierholm, S. R. Quartz, and L. Shams. Bayesian priors are encoded independently from likelihoods in human multisensory perception. *Journal of vision*, 9(5):23, 2009.

[14] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3, 2000.

[15] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden markov model hybrid. *Neural Networks, IEEE Transactions on*, 3(2):252–259, 1992.

[16] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[17] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[18] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[19] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, pages 1365–1372. IEEE, 2009.

[20] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[21] S. Boyd and L. Xiao. Least-squares covariance matrix adjustment. *SIAM Journal on Matrix Analysis and Applications*, 27(2):532–546, 2005.

[22] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[23] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *CVPR*, pages 26–33, 2003.

[24] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[25] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[26] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[27] 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.

[28] L. Cayton and S. Dasgupta. Robust euclidean embedding. In *Proceedings of the 23rd international conference on machine learning*, pages 169–176. ACM, 2006.

[29] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[30] Y.-T. Chen and C.-S. Chen. Fast human detection using a novel boosted cascading structure with meta stages. *Image Processing, IEEE Transactions on*, 17(8):1452–1464, 2008.

[31] T. F. Coleman and Y. Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.

[32] K. Crammer and Y. Singer. Improved output coding for classification using continuous relaxation. In *In Advances in Neural Information Processing Systes 13 (NIPS*00*, 2001.

[33] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[34] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *Machine Learning*, volume 47, pages 201–233, 2002.

[35] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886 –893 vol. 1, 2005.

[36] F. De la Torre. A least-squares framework for component analysis. *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, 34(6):1041–1055, 2012.

[37] F. De la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel. Detailed human data acquisition of kitchen activities: the cmu-multimodal activity database (cmu-mmac). Technical report, RI-TR-08-22h, CMU, 2008.

[38] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[39] 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.

[40] P. E. Downing, Y. Jiang, M. Shuman, and N. Kanwisher. A cortical area selective for visual processing of the human body. *Science*, 293(5539):2470–2473, 2001.

[41] 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.

[42] R. P. Duin and E. Pekalska. The science of pattern recognition. achievements and perspectives. In *Challenges for computational intelligence*, pages 221–259. Springer, 2007.

[43] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2179–2195, 2009.

[44] R. Epstein and N. Kanwisher. A cortical representation of the local visual environment. *Nature*, 392(6676):598–601, 1998.

[45] S. Escalera, A. Fornes, O. Pujol, P. Radeva, G. Sanchez, and J. Llados. Blurred shape model for binary and grey-level symbol recognition. *Pattern Recognition Letters*, 30:1424–1433, 2009.

[46] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. J. Escalante. Multi-modal gesture recognition challenge 2013: Dataset and results. *ChaLearn Multi-modal Gesture Recognition Grand Challenge and Workshop, 15th ACM International Conference on Multimodal Interaction*, pages 445–452, 2013.

[47] S. Escalera, O. Pujol, and P.Radeva. On the decoding process in ternary error-correcting output codes. *Transactions in Pattern Analysis and Machine Intelligence*, 99(1), 2009.

[48] S. Escalera, O. Pujol, and P. Radeva. Ecoc-one: A novel coding and decoding strategy. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 578 –581, 2006.

[49] S. Escalera, O. Pujol, and P. Radeva. Boosted landmarks and forest ECOC: A novel framework to detect and classify objects in clutter scenes. In *Pattern Recognition Letters*, volume 28, pages 1759–1768, 2007.

[50] S. Escalera, O. Pujol, and P. Radeva. On the decoding process in ternary error-correcting output codes. *PAMI*, 32:120–134, 2010.

[51] S. Escalera, D. M. Tax, O. Pujol, P. Radeva, and R. P. Duin. Subclass problem-dependent design for error-correcting output codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1041–1054, 2008.

[52] L. J. Eshelman. The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *FOGA*, pages 265–283, 1990.

[53] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[54] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[55] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[56] 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.

[57] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, pages 23–37, 1995.

[58] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[59] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[60] T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*, 2011.

[61] T. Gao and D. Koller. Multiclass boosting with hinge loss based on output coding. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 569–576, 2011.

[62] N. Garcia-Pedrajas and C. Fyfe. Evolving output codes for multiclass problems. *Evolutionary Computation, IEEE Transactions on*, 12(1):93 –106, 2008.

[63] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

[64] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[65] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations Research Letters*, 26(3):127 – 136, 2000.

[66] N. Guan, D. Tao, Z. Luo, and B. Yuan. Nenmf: an optimal gradient method for nonnegative matrix factorization. *Signal Processing, IEEE Transactions on*, 60(6):2882–2898, 2012.

[67] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014.

[68] A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov, and S. Escalera. Graph cuts optimization for multi-limb human segmentation in depth maps. In *CVPR*, pages 726–732, 2012.

[69] N. J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA journal of Numerical Analysis*, 22(3):329–343, 2002.

[70] J. Holland. *Adaptation in natural and artificial systems: An analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, 1975.

[71] C.-J. Hsieh and I. S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011.

[72] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification. *Department of CSIE, technical report*, 2002.

[73] F. J. Huang and Y. LeCun. Large-scale learning with svm and convolutional for generic object categorization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 284–291. IEEE, 2006.

[74] G. B. Huang, M. Ramesh, T. Berg, and E. L. Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report University of Massachusets Amherst, 07-49, October 2007.

[75] W. Hwang, J. Weng, and Y. Zhang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, 2003.

[76] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, 2000.

[77] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[78] N. Kanwisher, J. McDermott, and M. M. Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *The Journal of Neuroscience*, 17(11):4302–4311, 1997.

[79] H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008.

[80] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

[81] E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *ICML*, pages 313–321, 1995.

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[83] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[84] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

[85] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[86] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[87] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muiller, E. Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*, pages 53–60, 1995.

[88] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[89] A. C. Lorena and A. C. P. L. F. Carvalho. Evolutionary design of multiclass support vector machines. *J. Intell. Fuzzy Syst.*, 18:445–454, October 2007.

[90] A. C. Lorena and A. C. de Carvalho. Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomputing*, 71(16-18):3326 – 3334, 2008.

[91] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[92] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–1438, 2006.

[93] J. Malick. A dual approach to semidefinite least-squares problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):272–284, 2004.

[94] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *Computer Vision–ECCV 2008*, pages 479–491. Springer, 2008.

[95] A. Martinez and R. Benavente. The AR face database. In *Computer Vision Center Technical Report #24*, 1998.

[96] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *Knowledge and Data Engineering, IEEE Transactions on*, 20(10):1348–1362, 2008.

[97] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Computer Vision-ECCV 2004*, pages 69–82. Springer, 2004.

[98] I. Mukherjee and R. E. Schapire. A theory of multiclass boosting. *The Journal of Machine Learning Research*, 14(1):437–497, 2013.

[99] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[100] N. J. Nilsson. *Learning machines: foundations of trainable pattern-classifying systems.* McGraw-Hill, 1965.

[101] OSU-SVM-TOOLBOX. http://svm.sourceforge.net/.

[102] A. Passerini, M. Pontil, and P. Frasconi. New results on error correcting output codes of kernel machines. *Neural Networks, IEEE Transactions on*, 15(1):45–54, 2004.

[103] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Learning machines. In *McGraw-Hill*, 1965.

[104] R. Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.

[105] O. Pujol, S. Escalera, and P. Radeva. An incremental node embedding technique for error correcting output codes. *Pattern Recognition*, 41(2):713 – 725, 2008.

[106] 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.

[107] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[108] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2006.

[109] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1):65 –81, jan. 2007.

[110] P. Richtárik and M. Takávc. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[111] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *JMLR*, 5:101–141, 2004.

[112] D. L. Rohde. Methods for binary multidimensional scaling. *Neural Computation*, 14(5):1195–1232, 2002.

[113] F. Rosenblatt. Principles of neurodynamics. 1962.

[114] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004.

[115] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[116] M. J. Saberian and N. Vasconcelos. Multiclass boosting: Theory and algorithms. In *Advances in Neural Information Processing Systems*, pages 2124–2132, 2011.

[117] D. Sanchez, M. Á. Bautista, and S. Escalera. Hupba 8k+: Dataset and ecoc-graphcut based segmentation of human limbs. *Neurocomputing*, 2014.

[118] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 422–429. IEEE, 2010.

[119] R. E. Schapire. Using output codes to boost multiclass learning problems. In *ICML*, volume 97, pages 313–321, 1997.

[120] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[121] P. Simard, Y. LeCun, and J. S. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5, NIPS 1992*, pages 50–58. Morgan Kaufmann Publishers Inc., 1992.

[122] P. Simeone, C. Marrocco, and F. Tortorella. Design of reject rules for ecoc classification systems. *Pattern Recognition*, 45(2):863 – 875, 2012.

[123] D. F. Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

[124] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[125] T.Hastie and R.Tibshirani. Classification by pairwise grouping. *NIPS*, 26:451–471, 1998.

[126] D. Tran and D. Forsyth. Improved human parsing with a full relational model. In *Computer Vision–ECCV 2010*, pages 227–240. Springer, 2010.

[127] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[128] W. Utschick and W. Weichselberger. Stochastic organization of output codes in multiclass learning problems. In *Neural Computation*, volume 13, pages 1065–1102, 2004.

[129] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[130] V. Vineet, J. Warrell, L. Ladicky, and P. Torr. Human instance segmentation from video using detector-based conditional random fields. In *BMVC*, 2011.

[131] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, 2001.

[132] S. Watanabe. *Pattern Recognition: Human and Mechanical.* John Wiley & Sons, Inc., New York, NY, USA, 1985.

[133] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[134] T. H. Weisswange, C. A. Rothkopf, T. Rodemann, and J. Triesch. Bayesian cue integration as a developmental outcome of reward mediated learning. *PLoS One*, 6(7):e21575, 2011.

[135] J. Weston, C. Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

[136] T. Windeatt and R. Ghaderi. Coding and decoding for multi-class learning problems. In *Information Fusion*, volume 4, pages 11–21, 2003.

[137] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[138] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1385–1392. IEEE, 2011.

[139] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures-of-parts. 2012.

[140] B. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zöllei, J. R. Polimeni, et al. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of neurophysiology*, 106(3):1125–1165, 2011.

[141] X. Zhang, L. Liang, and H.-Y. Shum. Spectral error correcting output codes for efficient multiclass recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1111–1118, Sept 2009.

[142] B. Zhao and E. P. Xing. Sparse output coding for large-scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3350–3357. IEEE, 2013.

[143] G. Zhong and M. Cheriet. Adaptive error-correcting output codes. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1932–1938. AAAI Press, 2013.

[144] G. Zhong, K. Huang, and C.-L. Liu. Joint learning of error-correcting output codes and dichotomizers from data. *Neural Computing and Applications*, 21(4):715–724, 2012.

[145] J. Zhou, H. Peng, and C. Y. Suen. Data-driven decomposition for multi-class classification. *Pattern Recognition*, 41(1):67 – 76, 2008.

[146] J. D. Zhou, X. D. Wang, H. J. Zhou, J. M. Zhang, and N. Jia. Decoding design based on posterior probabilities in ternary error-correcting output codes. *Pattern Recognition*, 45(4):1802 – 1818, 2012.

[147] J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. *Statistics and Its*, 2009.

[148] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.