

Error-Correcting Factorization

Miguel Angel Bautista, Oriol Pujol, Fernando De la Torre and Sergio Escalera

arXiv:1502.07976v2 [cs.CV] 5 Mar 2015

Abstract—Error Correcting Output Codes (ECOC) is a successful technique in multi-class classification, which is a core problem in Pattern Recognition and Machine Learning. A major advantage of ECOC over other methods is that the multi-class problem is decoupled into a set of binary problems that are solved independently. However, literature defines a general error-correcting capability for ECOCs without analyzing how it distributes among classes, hindering a deeper analysis of pairwise error-correction. To address these limitations this paper proposes an Error-Correcting Factorization (ECF) method, our contribution is three fold: (I) We propose a novel representation of the error-correction capability, called the design matrix, that enables us to build an ECOC on the basis of allocating correction to pairs of classes. (II) We derive the optimal code length of an ECOC using rank properties of the design matrix. (III) ECF is formulated as a discrete optimization problem, and a relaxed solution is found using an efficient constrained block coordinate descent approach. (IV) Enabled by the flexibility introduced with the design matrix we propose to allocate the error-correction on classes that are prone to confusion. Experimental results in several databases show that when allocating the error-correction to confusable classes ECF outperforms state-of-the-art approaches.

Index Terms—Error-Correcting Output Codes, Multi-class learning, Matrix Factorization

1 INTRODUCTION

In the last decade datasets have experimented an exponential growth rate, generating vast collections of data that need to be automatically analyzed. In particular, multimedia datasets have experienced an explosion on data availability, thanks to the almost negligible cost of gathering multi-media data from Internet. Therefore, there is a pushing need for efficient algorithms that are able to automatize knowledge extraction processes on those datasets. One of the classic problems in Pattern Recognition and Machine Intelligence is to perform automatic classification, i.e., automatically attributing a label to each sample of the dataset. In this sense, the classification process is often considered as first step for higher order representations or knowledge extractions. In multi-class classification problems the goal is to find a function $f : \mathbb{R}^n \rightarrow \mathbb{K}$, that maps samples to a finite discrete set \mathbb{K} of labels with $|\mathbb{K}| > 2$. While there exists a large set of approaches to estimate f all of them can be grouped in two different categories: *Single-Machine/Single-Loss* approaches and *Divide and Conquer*

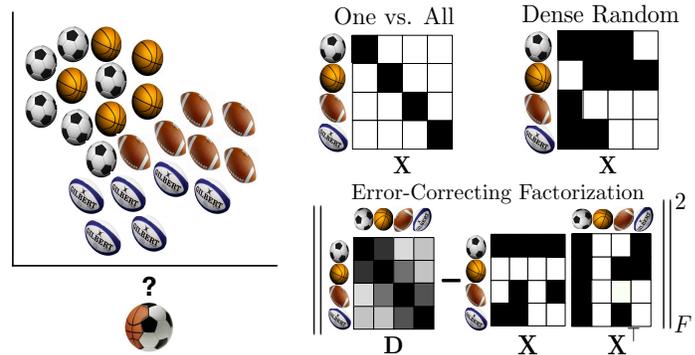


Fig. 1. Example of a classification problem of 4 different sports balls. Note how *One vs. All* or *Dense Random* ECOC designs do not take into account the data distribution while the proposed Error-Correcting Factorization method finds an ECOC matrix X by factorizing a design matrix D . In addition, the codes (rows of X) ECF assigns to similar categories are very dissimilar in order to benefit from Error-Correcting principles.

approaches. The formers attempt to approximate a single f for the complete multi-class problem, while the latter decouple f into a set of binary sub-functions (binary classifiers) that are potentially easier to estimate and aggregate the results.

In this sense, Error-Correcting Output Codes (ECOC) is a divide and conquer approach that has proven to be very effective in many different multi-class contexts. The core property within an ECOC is its capability to correct errors in binary classifiers by using redundancy. However, existing literature represents the error-correcting capability of an ECOC as a scalar, hindering a deeper the analysis of error-correction and redundancy on class pairs. Furthermore, classical divide and conquer approaches that have been included in the ECOC framework like *One vs. All* [48] or *Random* [2] approaches ignore the data distribution, thus not taking profit of allocating the error-correcting capabilities of ECOCs in a problem-dependent fashion. In addition, recent problem-dependent ECOC designs have focused on designing the binary sub-functions rather than analyzing the core error-correcting property. In order to overcome this limitations, our proposal builds an ECOC matrix by factorizing a design matrix D that encodes the desired ‘correction properties’ between classes (i.e a design matrix which can be obtained directly from data or be set by experts on the problem domain). The proposed method finds the ECOC coding that yields the closest configuration to the design matrix. We cast the task of designing an ECOC as a matrix factorization problem with binary constraints. A visual example is shown in Figure 1.

- Miguel Angel Bautista, Oriol Pujol and Sergio Escalera are with the Department of Applied Mathematics and Analysis, University of Barcelona, Barcelona, 08007, Spain and the Computer Vision Center, Autonomous University of Barcelona, 08193, Bellaterra (Cerdanyola), Barcelona, Spain E-mail: {mbautista,sescalera,oriol_pujol}@ub.edu
- Fernando De la Torre is in the Robotics Institute at Carnegie Mellon University, Pittsburgh, 15216, PA. E-mail: ftorre@cs.cmu.edu

2 RELATED WORK

2.1 Single-machine/Single-loss Approaches

The multi-class problem can be directly treated by some methods that exhibit a multi-class behaviour off the shelf (i.e. Nearest Neighbours [22], Decision Trees [30], Random Forests [6]). 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 [7] to the multi-class case. For instance, the work of Weston and Watkins [55] 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 [14] 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 [48], [52]. In recent years various works that extended the classical Adaptive Boosting method [20] to the multi-class setting have been presented [51], [43]. In [62] 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 Vasconcelos [50] 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.

2.2 Divide and Conquer Approaches

On the other hand, the divide and conquer approach has drawn a lot of attention due to its excellent results and easily parallelizable architecture [48], [52], [2], [18], [46], [4], [40], [28]. 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. 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 [52] and One vs. All [48] are those that use a predefined problem partition scheme followed by a committee strategy to aggregate the binary classifier outputs. On the other hand, hierarchical classification relies on a similarity metric distance

among classes to build a binary tree in which nodes correspond to different problem partitions [23], [40], [28]. 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 [16] (univocally defined, one code per class) (see Figure 2). 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 [17], [61]. The coding step has been widely studied in literature, yielding three different types of codings: predefined codings [48], [52], random codings [2] and problem-dependent codings for ECOC [18], [46], [4], [57], [24], [58]. Predefined codings like One vs. All or One vs. One are directly embeddable in the ECOC framework. In [2], the authors propose the Dense and Sparse Random coding designs with a fixed code length of $\{10, 15\} \log_2(K)$, respectively. In [2] 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.

2.3 Problem-dependent Strategies

Alternatively, problem-dependent strategies for ECOC have proven to be successful in multi-class classification tasks [57], [23], [24], [58], [18], [60], [59], [46]. 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, [57] 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 [24], 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 [23] 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. Finally, the work of Zhao et. al [58] 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.

2.4 Our approach

In this paper we present the Error-Correcting Factorization (ECF) method for factorizing a design matrix 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

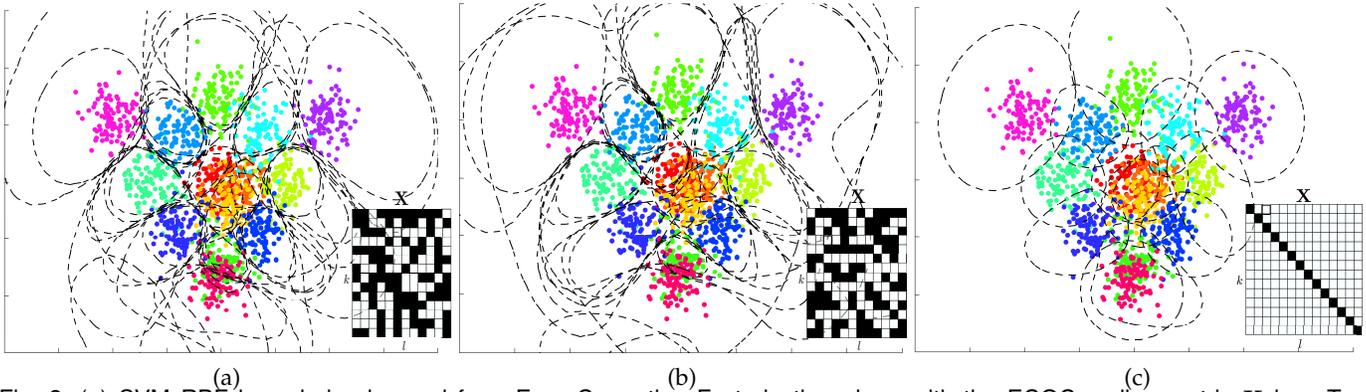


Fig. 2. (a) SVM RBF boundaries learned from Error-Correcting Factorization along with the ECOC coding matrix \mathbf{X} in a Toy problem, 77.12% classification accuracy (12 classifiers are trained). (b) Boundaries learned by the Dense Random ECOC coding design, 66.45% classification accuracy (12 classifiers are trained). (c) SVM boundaries induced by the One vs. All approach, 49.53% classification accuracy (14 classifiers are trained).

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. Figure 2 shows different coding schemes and the real boundaries learned by binary classifiers (SVM with RBF kernel) for a Toy problem of 14 classes (see section 5 for further details on the dataset). We can see how the binary problems induced by ECF in Fig. 2(a) boost the boundaries of classes that are prone to be confused, while other approaches that use equal or higher number of classifiers like Dense Random [2] in Fig. 2(b), or classic One vs. All designs in Fig. 2(c) fail in this task. The paper is organized as follows: Section 3 introduces the ECOC properties and derives ECF, where we cast the problem of finding an ECOC matrix that follows a certain distribution of correction as a discrete optimization problem. Section 4 presents a discussion of the method addressing important issues from the point of view of the ECOC framework. Concretely, 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. Section 5 shows how ECF yields ECOC coding matrices that obtain higher classification performances than state-of-the-art methods with comparable or lower computational complexity. Finally, Section 6 concludes the paper.

3 METHODOLOGY

In this section, we review existing properties of the ECOC framework and propose to cast the ECOC coding matrix optimization as a Matrix Factorization problem that can be solved efficiently using a constrained coordinate descent approach.

3.1 Error-Correcting Output Codes

ECOC is a multi-class framework inspired on the basis of error-correcting principles of communication theory [16], which is composed of two different steps: *coding* [16], [2] and *decoding* [17], [61]. At the coding step an ECOC coding matrix $\mathbf{X} \in \{-1, +1\}^{k \times l}$ (see notation¹) is constructed, where k denotes the number of classes in the problem and l the number of bi-partitions (also known as dichotomies) to be learnt. In the coding matrix, the rows (\mathbf{x}^i 's, also known as *codewords*) are unequivocally defined, since these are the identifiers of each category in the multi-class problem. On the other hand, the columns of \mathbf{X} (\mathbf{x}_j 's) denote the bi-partitions to be learnt by base classifiers (also known as dichotomizer). Therefore, for a certain column a dichotomizer learns the boundary between classes valued $+1$ and classes valued -1 . However, [2] introduced a third value, defining ternary valued coding matrices. $\mathbf{X} \in \{-1, +1, 0\}^{k \times l}$. In this case, for any 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 as One vs. One [52].

At the decoding step a data sample \mathbf{s} is classified among the $\{c_1, \dots, c_k\}$ possible categories. In order to perform the classification task, each dichotomizer predicts a binary value for \mathbf{s} whether it belongs to one of the bi-partitions defined by the correspondent dichotomy. Once the set of predictions $\mathbf{y} \in \{-1, +1\}^l$ is obtained, it is compared to the rows of \mathbf{X} using a distance function δ , known as the *decoding function*. Usual decoding techniques are based on well-known distance measures such as the l_1 or Euclidean distance. These measures are proved to be effective for $\mathbf{X} \in \{+1, -1\}^{k \times l}$. Nevertheless, it is not until the work of [17] that decoding functions took into account the meaning of the 0 value at the decoding step. Generally, the final prediction for \mathbf{s} is given by the class c_i , where $\arg \min_i \delta(\mathbf{x}^i, \mathbf{y})$, $i \in \{1, \dots, k\}$.

1. 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. x_{ij} denotes the scalar in the i -th row and j -th column of \mathbf{X} . $\|\mathbf{X}\|_F = \text{tr}(\mathbf{X}^T \mathbf{X})$ denotes the Frobenius norm. $\|\cdot\|_p$ is used to denote the L_p -norm. $\mathbf{x} \oplus \mathbf{y}$ is an operator which concatenates vectors \mathbf{x} and \mathbf{y} . $\text{rank}(\mathbf{X})$ denotes the rank of \mathbf{X} . $\mathbf{X} \leq 0$ denotes the point-wise inequality

3.2 Good practices in ECOC

Several works have studied the characteristics of a good ECOC coding matrix [16], [36], [3], [57], [4], which are summed up in the following three properties:

- 1) **Correction capability:** let $\mathbf{H} \in \mathbb{R}^{k \times k}$ denote a symmetric matrix of hamming distances among all pairs of rows in \mathbf{X} , the correction capability is expressed as $\lfloor \frac{\min(\mathbf{H})-1}{2} \rfloor^2$, considering only off-diagonal values of \mathbf{H} . In this sense, if $\min(\mathbf{H}) = 3$, 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.³
- 2) **Uncorrelated binary sub-problems:** the induced binary problems should be as uncorrelated as possible for \mathbf{X} to recover binary classifier errors.
- 3) **Use of powerful binary classifiers:** since the final class prediction consists of the aggregation of bit predictors, accurate binary classifiers are also required to obtain accurate multi-class predictions.

3.3 From global to pair-wise correction capability

In literature, correction capability has been a core objective of problem-dependent designs of \mathbf{X} . In this sense, different authors have always agreed on defining correction capability for an ECOC coding matrix as a global value [16], [2], [36], [57], [23], [25]. Hence, $\min(\mathbf{H})$ is expected to be large in order for \mathbf{X} to recover from as many binary classifier errors as possible. However, since \mathbf{H} expresses the hamming distance between rows of \mathbf{X} , one can alternatively express the correction capability in a pair-wise fashion [5], allowing for a deeper understanding of how correction is distributed among codewords. Figure 3 shows an example of global and pair-wise correction capabilities calculation. Recall that the \oplus operator between two vectors denotes its concatenation. Thus, the pair-wise correction capability is defined as follows:

- 4) The **pair-wise correction capability** of codewords \mathbf{x}^i and \mathbf{x}^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 the global correction capability of \mathbf{X} is 0, there are pairs of codewords with a higher correction, e.g. \mathbf{x}^2 and \mathbf{x}^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. In this sense, it is straightforward to demand the correction capabilities of the ECOC matrix to 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). However, recent works [57], [23], [58] have focused on designing a matrix \mathbf{X} where binary problems are easily separable. This assumption leads to a matrix \mathbf{X} where classes that are not easily separable

2. In the case of ternary codes this correction capability can be easily adapted.

3. Note that for \mathbf{X} to be valid all off-diagonal elements of \mathbf{H} should be greater or equal than one.

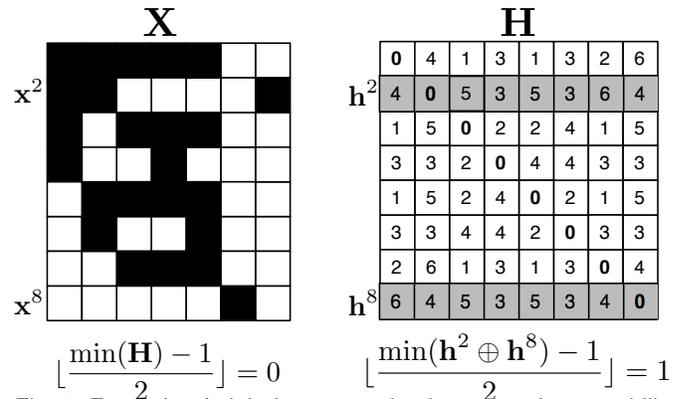


Fig. 3. Example of global versus pair-wise correction capability. On the left side of the figure the calculation of the global correction capability is shown. The right side of the image shows a sample of pair-wise correction calculation for codewords \mathbf{x}^2 and \mathbf{x}^8 .

show a small hamming distance on their respective codewords (i.e. following a "hard classes are left behind" scheme).

In addition to the proposal of a general method for ECOC coding by means of the definition of a design matrix, we explore the effect of focusing the learning effort of our method in those classes that have complex boundaries (i.e. those which show a small inter-class margin). It is important to take into account that though it is natural to estimate the design matrix from training data, it is not a limitation of ECF. In this sense, the design matrix can also code information of experts or any other distance measure directly set by the user. Formally, let $\mathbf{X} \in \{-1, +1\}^{k \times l}$ be a coding matrix, let \mathbf{H} be a symmetric matrix of pair-wise l_1 distances between rows of \mathbf{X} and let $\mathbf{D} \in \mathbb{R}^{k \times k}$ be a design matrix (e.g. pair-wise distance measure between class codewords). It is natural to see that the ordinal properties of the distance should hold in \mathbf{H} and \mathbf{D} . Thus, if distance between codewords \mathbf{x}^k and \mathbf{x}^l (d_{kl}) is required to be larger than the distance between codewords \mathbf{x}^i and \mathbf{x}^j (d_{ij}), this order should be maintained in \mathbf{H} . Then we want to find a configuration of \mathbf{X} such that $h_{ij} < h_{kl} \iff d_{ij} < d_{kl} \forall i, j, k, l$.

Note that the l_1 distances in \mathbf{H} can be seen as a function of the dot product of the codewords $\|\mathbf{x}^i - \mathbf{x}^j\|_1 = \frac{-(\mathbf{x}^i \mathbf{x}^j)^T + l}{2}$, where $\mathbf{x} \in \{-1, +1\}$. Therefore, instead of directly requiring \mathbf{H} to match \mathbf{D} , we can equivalently require the product $\mathbf{X}\mathbf{X}^T$ to match \mathbf{D} [54]. This implies that we can cast the problem of finding \mathbf{X} into a Matrix Factorization problem, where we find an \mathbf{X} so that the matrix of inner products $\mathbf{X}\mathbf{X}^T$ is closest to \mathbf{D} under a given norm.

3.4 Error-Correcting Factorization

This section describes the objective function and the optimization strategy for the ECF algorithm.

3.4.1 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{X} . This factorization is formulated as the quadratic form $\mathbf{X}\mathbf{X}^T$ that reconstructs \mathbf{D} with minimal Frobenius distance under several constraints, as shown in

Equation (1) ⁴.

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{D} - \mathbf{X}\mathbf{X}^\top\|_F^2 \quad (1)$$

$$\text{subject to} \quad \mathbf{X} \in \{-1, +1\}^{k \times l} \quad (2)$$

$$\mathbf{X}\mathbf{X}^\top - \mathbf{P} \leq 0 \quad (3)$$

$$\mathbf{X}^\top \mathbf{X} - \mathbf{1}(l-1) \leq 0 \quad (4)$$

$$-\mathbf{X}^\top \mathbf{X} - \mathbf{1}(l-1) \leq 0 \quad (5)$$

The component $\mathbf{X}^* \in \{-1, +1\}^{k \times l}$ 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{X} to be a valid matrix under the ECOC framework we constraint \mathbf{X} in Equations (2)-(5). Equation (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{X} , the constraints in 3 ensure that the correlation between rows of \mathbf{X} less or equal than a certain user-defined matrix $-1l \leq \mathbf{P} \leq 1l$ (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{x}^i and \mathbf{x}^j to be $\lfloor \frac{c-1}{2} \rfloor$, we set $p^i = p^j = \mathbf{1}(l-c)$.

Finally, constraints in Equations (4) and (5) ensure the induced binary problems are not equivalent. Similar constraints have been studied thoroughly in literature [16], [36], [25] defining methods that rely on diversity measures for binary problems to obtain a coding matrix \mathbf{X} . Equations (4) and (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 $\mathbf{X} \in \{-1, +1\}^{k \times l}$ that induces some equivalent binary problems but ensures that $\mathbf{X}\mathbf{X}^\top \leq \mathbf{1}(l-1)$, $\forall i, j : i \neq j$ will define a matrix whose rows are unequivocally defined. In this sense, a coding matrix \mathbf{X} can be easily projected on the set defined by constraints (4) and (5) by eliminating repeated columns, $\mathbf{X} = \mathbf{x}_j : \mathbf{x}_j \neq \mathbf{x}_i \forall j \neq i$. Thus, constraints in 4 and 5 ensure that uncorrelated binary sub-problems will be defined in our coding matrix \mathbf{X} . The discrete constraint in Equation 2 on the variable elevates the optimization problem to the NP-Hard class. To overcome this issue and following [13], [58], [8] we relax the discrete constraint in 2 an replace it by $\mathbf{X} \in [-1, +1]^{k \times l}$ in Equation 7.

3.4.2 Optimization

In this section, we detail the process for optimizing \mathbf{X} . The minimization problem posed in Equation (1) with the relaxation of the boolean constraint in Equation (2) is non-convex, thus, \mathbf{X}^* 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 [49], [35], [1] these techniques do not enjoy the efficiency and scalability properties present in other optimization methods applied to Matrix Factorization problems, such as Coordinate Descent [37], [15]. Coordinate Descent techniques have been widely applied in Nonnegative Matrix Factorization obtaining satisfying results in terms of efficiency [34], [31]. In addition, it has

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

been proved that if each of the coordinate sub-problems can be solved exactly, Coordinate Descent converges to a stationary point [29], [53]. Using this result, we decouple the problem in Equation (1) into a set of linear least-squares problems (one for each coordinate). Therefore, if the problem in Equation (1) is going to be minimized along the i -th coordinate of \mathbf{X} , we fix all rows of \mathbf{X} except of \mathbf{x}^i and we substitute \mathbf{X} with $\begin{bmatrix} \mathbf{x}^i \\ \mathbf{X}'^i \end{bmatrix}$ in Equations (1) and (3), where \mathbf{X}'^i denotes matrix \mathbf{X} after removing the i -th row. In addition, we substitute \mathbf{D} with $\begin{bmatrix} l & \mathbf{d}_i \\ \mathbf{d}_i{}^T & \mathbf{D}'^i \end{bmatrix}$, where \mathbf{D}'^i denotes the matrix \mathbf{D} after removing the i -th row and column. Equivalently, we substitute $\mathbf{P} = \begin{bmatrix} l & \mathbf{p}_i \\ \mathbf{p}_i{}^T & \mathbf{P}'^i \end{bmatrix}$, obtaining the following block decomposition:

$$\underset{\mathbf{x}^i}{\text{minimize}} \quad \left\| \begin{bmatrix} l & \mathbf{d}_i \\ \mathbf{d}_i{}^T & \mathbf{D}'^i \end{bmatrix} - \begin{bmatrix} \mathbf{x}^i \mathbf{x}^i{}^T & \mathbf{X}'^i \mathbf{x}^i \\ \mathbf{X}'^i \mathbf{x}^i{}^T & \mathbf{X}'^i \mathbf{X}'^i{}^\top \end{bmatrix} \right\|_F^2 \quad (6)$$

$$\text{subject to} \quad \mathbf{x}^i \in [-1, +1]^l \quad (7)$$

$$\begin{bmatrix} \mathbf{x}^i \mathbf{x}^i{}^T & \mathbf{X}'^i \mathbf{x}^i \\ \mathbf{X}'^i \mathbf{x}^i{}^T & \mathbf{X}'^i \mathbf{X}'^i{}^\top \end{bmatrix} - \begin{bmatrix} l & \mathbf{p}_i \\ \mathbf{p}_i{}^T & \mathbf{P}'^i \end{bmatrix} \leq 0. \quad (8)$$

Analyzing the block decomposition in Equation (6) we can see that the only terms involving free variables are $\mathbf{x}^i \mathbf{x}^i{}^T$, $\mathbf{X}'^i \mathbf{x}^i$ and $\mathbf{X}'^i \mathbf{x}^i{}^T$. Thus, since \mathbf{D} and $\mathbf{X}\mathbf{X}^\top$ are symmetric by definition, the minimizer \mathbf{x}^{i*} of Equation (6) is the solution to the linear least-squares problem shown in Equation (9):

$$\underset{\mathbf{x}^i}{\text{minimize}} \quad \left\| \mathbf{X}'^i \mathbf{x}^i - \mathbf{d}^i \right\|_2^2 \quad (9)$$

$$\text{subject to} \quad -1 \leq \mathbf{x}^i \leq +1 \quad (10)$$

$$\mathbf{X}'^i \mathbf{x}^i - \mathbf{p}^i \leq 0, \quad (11)$$

where constraint (10) is the relaxation of the discrete constraint (2). In addition, constraint (11) ensures the correlation of \mathbf{x}^i with the rest of the rows of \mathbf{X} is below a certain value p^i . Algorithm 1 shows the complete optimization process.

Algorithm 1: Error-Correcting Factorization Algorithm.

Data: $\tilde{\mathbf{D}} \in \mathbb{R}^{k \times k}$, $\mathbf{P} \in \mathbb{N}^{k \times k}$, l

Result: $\mathbf{X} \in \{-1, +1\}^{k \times l}$

begin

repeat

foreach $i \in \{1, 2, \dots, k\}$ **do**

$\mathbf{x}^i \leftarrow \underset{\mathbf{x}^i \in \mathbb{R}^l}{\text{minimize}} \left\| \mathbf{X}'^i \mathbf{x}^i - \mathbf{d}^i \right\|_2^2$, subject to :

$-1 \leq \mathbf{x}^i \leq +1$, $\mathbf{X}'^i \mathbf{x}^i - \mathbf{p}^i \leq 0$;

$\mathbf{X} \leftarrow \epsilon$ -suboptimal(\mathbf{X});

$\mathbf{X} = \{\mathbf{x}_j : \mathbf{x}_j \neq \mathbf{x}_i \forall j \neq i\}$; // Projection step
 to remove duplicate columns

until convergence;

To solve the minimization problem in Algorithm 1 we use the Active Set method described in [26], which finds an initial feasible solution by first solving a linear programming problem. Once ECF converges to a solution \mathbf{X}^* with objective value $f_{obj}(\mathbf{X}^*)$ we obtain a discretized ϵ -suboptimal solution $\mathbf{X} \in \{-1, +1\}$ with objective value $f_{obj}(\mathbf{X})$ by sampling 1000 points that split the interval $[-1, +1]$ and choosing the point that minimizes $\|f_{obj}(\mathbf{X}^*) - f_{obj}(\mathbf{X})\|_2$. Finally, we discard

repeated columns if any appear ⁵.

3.5 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 (1) are found in other contexts, for example, in the Singular Value Decomposition problem (SVD). The SVD uses the same objective function as ECF subjected to the constraint $\mathbf{X}\mathbf{X}^\top = \mathbf{I}$. However, the solution of SVD yields an orthogonal basis, disagreeing with the objective defined in Equation (1) which ensures different correlations between the \mathbf{x}^i 's. In addition, we can also find a common ground with the Nearest Correlation Matrix (NMC) Problem [32], [9], [39]. However, the NMC solution does not yield a discrete factor \mathbf{X} , instead it seeks directly for the Gramian $\mathbf{X}\mathbf{X}^\top$ where \mathbf{X} is not discrete, as in Equation (12).

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{D}\|_F^2 \quad (12)$$

$$\text{subject to} \quad \mathbf{X} \succeq 0 \quad (13)$$

$$\mathbf{c}\mathbf{X}\mathbf{c}^\top = \mathbf{b} \quad (14)$$

In addition, the ECF has similarities with the Discrete Basis Problem (DBP) [42], since the factors are \mathbf{X} 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 (15).

$$\underset{\mathbf{X}, \mathbf{Y}}{\text{minimize}} \quad \|\mathbf{X} \circ \mathbf{Y} - \mathbf{D}\|_1 \quad (15)$$

$$\text{subject to} \quad \mathbf{X}, \mathbf{Y}, \mathbf{D} \in \{0, 1\} \quad (16)$$

4 DISCUSSION

In this section we discuss how to ensure that the design matrix \mathbf{D} is valid, as well as how to automatically estimate the code length for each problem given \mathbf{D} . Furthermore, we analyze the convergence of ECF in relation to the order of updating the coordinates. Finally we show that under certain conditions of \mathbf{D} ECF converges to a solution with almost negligible objective value.

4.1 Ensuring a representable design matrix

An alternative interpretation for ECF is that it seeks for a discrete matrix \mathbf{X} 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 [32], [9], [11], [27], 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 $\tilde{\delta}_{ii} = lVi$. In this sense, to find $\tilde{\mathbf{D}}$ we follow an alternating projections algorithm, similar as [32], which is shown in Algorithm 2. We first project \mathbf{D} into the PSD cone by computing its eigenvectors and recovering $\mathbf{D} = \mathbf{V} \text{diag}(\boldsymbol{\lambda}_+) \mathbf{V}^\top$, where $\boldsymbol{\lambda}_+$ are the non-negative eigenvalues of \mathbf{D} . Then, we scale \mathbf{D} in the range $[-l, +l]$ and set $\delta_{ii} = lVi$.

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

Algorithm 2: Projecting \mathbf{D} into the PSD cone with additional constraints.

Data: $\mathbf{D} \in \mathbb{R}^{k \times k}$

Result: $\tilde{\mathbf{D}} \in \mathbb{R}^{k \times k}$

begin

repeat

$\mathbf{D} \leftarrow \mathbf{V} \text{diag}(\boldsymbol{\lambda}_+) \mathbf{V}^\top$;

$\mathbf{D} \leftarrow \mathbf{D} \in [-l, +l]^{k \times k}$;

$\mathbf{D} \leftarrow d_{ii} = lVi$;

until convergence;

4.2 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 [57], [58], [4], [18]) are suggested [2] 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 [57] 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 [38] 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 $\mathbf{X}\mathbf{X}^\top$, there are three different possibilities:

- 1) If $\text{rank}(\mathbf{X}\mathbf{X}^\top) = \text{rank}(\mathbf{D})$, we obtain rank factorization algorithm that should be able to factorize \mathbf{D} with minimal error.
- 2) In the case when $\text{rank}(\mathbf{X}\mathbf{X}^\top) < \text{rank}(\mathbf{D})$ 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 $\text{rank}(\mathbf{X}\mathbf{X}^\top) > \text{rank}(\mathbf{D})$, the system is overdetermined and many possible solutions exist.

In general we would like to reconstruct \mathbf{D} with minimal error, and since $\text{rank}(\mathbf{X}) \leq \min(k, l)$ and k (the number of classes) is fixed, we only have to set the number of columns of \mathbf{X} to control the rank. Hence, by setting $\text{rank}(\mathbf{X}) = l = \text{rank}(\mathbf{D})$, ECF will be able to factorize \mathbf{D} with minimal error. Figure 4 shows visual results for the ECF method applied 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 4(e)(f).

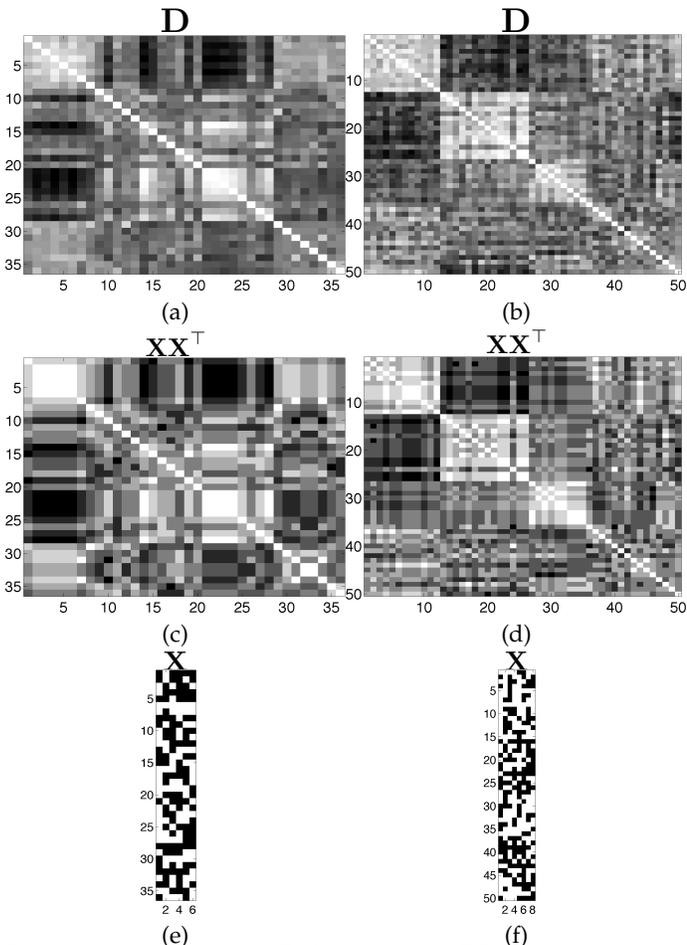


Fig. 4. \mathbf{D} matrix for the *Traffic* (a) and *ARFace* (b) datasets. \mathbf{XX}^T term obtained via ECF for *Traffic* (c) and *ARFace* (d) datasets. ECOC coding matrix \mathbf{X} obtained with ECF for *Traffic* (e) and *ARFace* (f).

4.3 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 [47], [21], [53], [33]. In particular, [44] 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 is a suitable choice when the problem shows some of the following characteristics [47]:

- 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 [44].

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 lead us to choose a cyclic update scheme for ECF. In addition in Figure 5 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 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.

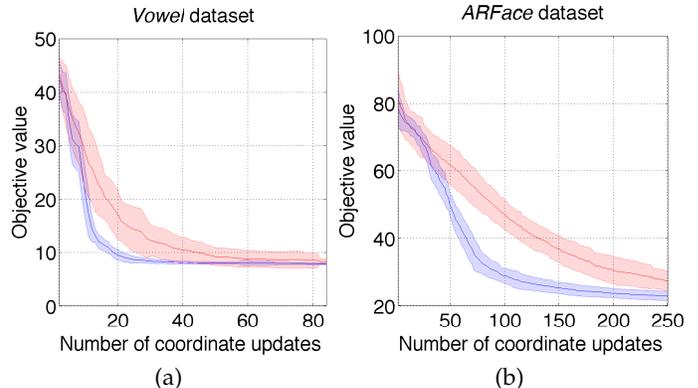


Fig. 5. Mean Frobenius norm value with standard deviation as a function of the number of coordinate updates on 50 different trials. The blue shaded area corresponds to cyclic update while the red area denotes random coordinate updates for *Vowel* (a) and *ARFace* (b) datasets.

4.4 Approximation Errors and Convergence results when \mathbf{D} is an inner product of binary data

The optimization problem posed by ECF in Equation (1) is non-convex due to the quadratic term \mathbf{XX}^T , 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{XX}^T that is closest to \mathbf{D} under the Frobenius norm. Hence, the error in the approximation can be measured by $\|\mathbf{X}^*\mathbf{X}^{*\top} - \mathbf{D}\|_F^2 \geq 0$, where \mathbf{X}^* 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 Frobenius norm. Thus, we expand the norm as in the following equation:

$$\|\mathbf{X}^*\mathbf{X}^{*\top} - \mathbf{D}\|_F^2 = \|\mathbf{X}^*\mathbf{X}^{*\top} - \mathbf{D}^B + \mathbf{D}^B - \mathbf{D}\|_F^2 = \quad (17)$$

$$= \|\mathbf{X}^*\mathbf{X}^{*\top} - \mathbf{D}^B\|_F^2 + \|\mathbf{D} - \mathbf{D}^B\|_F^2 - \quad (18)$$

$$- 2\text{tr}((\mathbf{X}^*\mathbf{X}^{*\top} - \mathbf{D}^B)(\mathbf{D} - \mathbf{D}^B)). \quad (19)$$

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

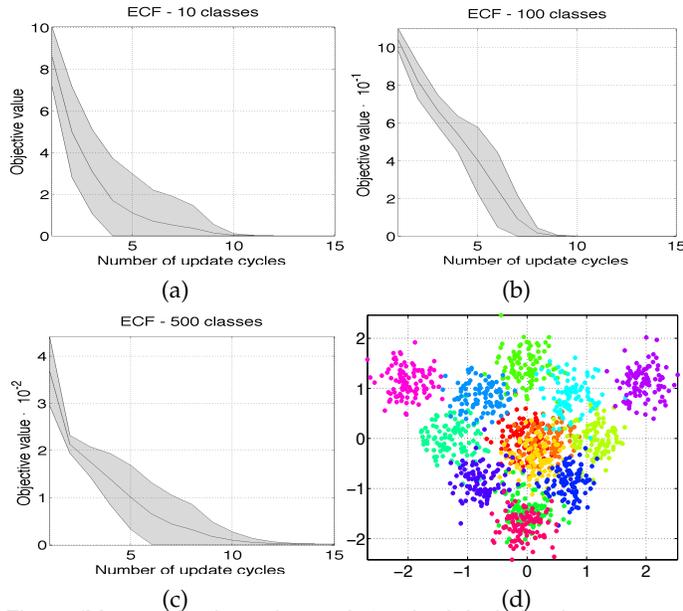


Fig. 6. (Mean objective value and standard deviation for 30 runs of ECF on a random D^G of 10 classes (a), 100 classes (b), and 500 classes (c). (d) Toy problem synthetic data, where each color corresponds to a different category in the multi-class problem.

of different sizes and calculate the average $\bar{\epsilon}_o$. Figure 6 shows examples for different D^G matrices of size 10×10 , 100×100 , and 500×500 . In Figure 6 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 D^{B^i} s after 15 update cycles (coordinate updates for all x^i 's) is $\bar{\epsilon}_o < 10^{-10}$. This implies, that ECF converges in average to a point with almost negligible objective value, and when applied to 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 D this discretization error is unavoidable, and as we have seen empirically, ECF converges in average to a solution with almost negligible objective value.

5 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.

5.1 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 6(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

TABLE 1
Dataset characteristics.

	Class	Segment.	Ecoli	Yeast	Vowel	Toy	Traffic	ARFace
#s	214	2310	336	1484	990	400	3481	1300
#f	9	19	8	8	10	2	100	120
#c	7	7	8	10	11	14	36	50



Fig. 7. Visual examples for the *ARFace* and *Traffic* datasets.

from the ARFaces [41] dataset are classified using the present methodology. These datasets are public upon request to the authors. Table 1 shows the characteristics of the different datasets.

•**Traffic sign categorization:** We test ECF on a real traffic sign categorization problem, of 36 classes [10]. 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. A visual sample is show in Figure 7(a).

•**ARFaces classification:** The ARFace database [41] is composed of 26 face images from 126 different subjects (from which 50 are selected), portraying different expressions and complements. An example is shown in Figure 7(b).

5.2 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 [48], [52]. 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 [3] (*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 [57] and the Relaxed Hierarchy [23] (*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 [23], [58], [57], 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 tuned 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 [12]. 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 [23] 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 [23]) we used both the Hamming Decoding (HD) and the Loss-Weighted decoding (LWD) [46].

5.3 Experimental Results

In Figure 8 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 [23]. 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 9 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 8 and 9 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. 8(e)-(f) and 9(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 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.

TABLE 2

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

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 10 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 $\pm 5\%$ 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 10(a) and 10(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 10(b) and 10(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 10(c) and 10(f) show the comparison for *OVA*, which is a standard method often defended by its simplicity [48]. We clearly see how it never outperforms any method and it is not the recommended choice for almost any dataset. In Table 2 we show the percentage of wins for all methods⁶, 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 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.

6 CONCLUSIONS

We presented the Error-Correcting Factorization method for multi-class learning which is based on the Error-Correcting Output Codes framework. The proposed method 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

6. The R-H method [23] 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.

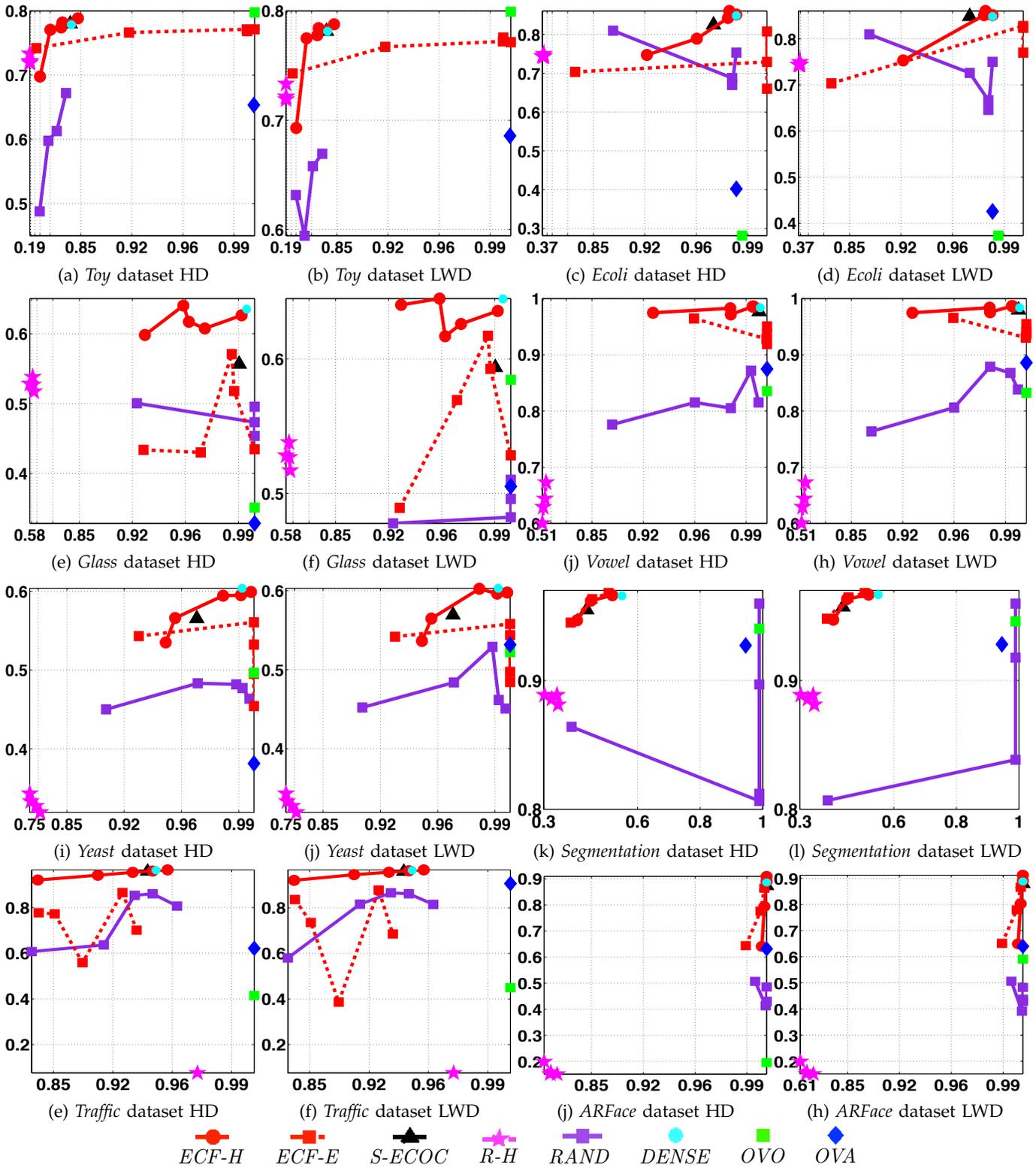


Fig. 8. Multi-class classification accuracy (y axis) as a function of the relative computational complexity (x axis) for all datasets and both decoding measures.

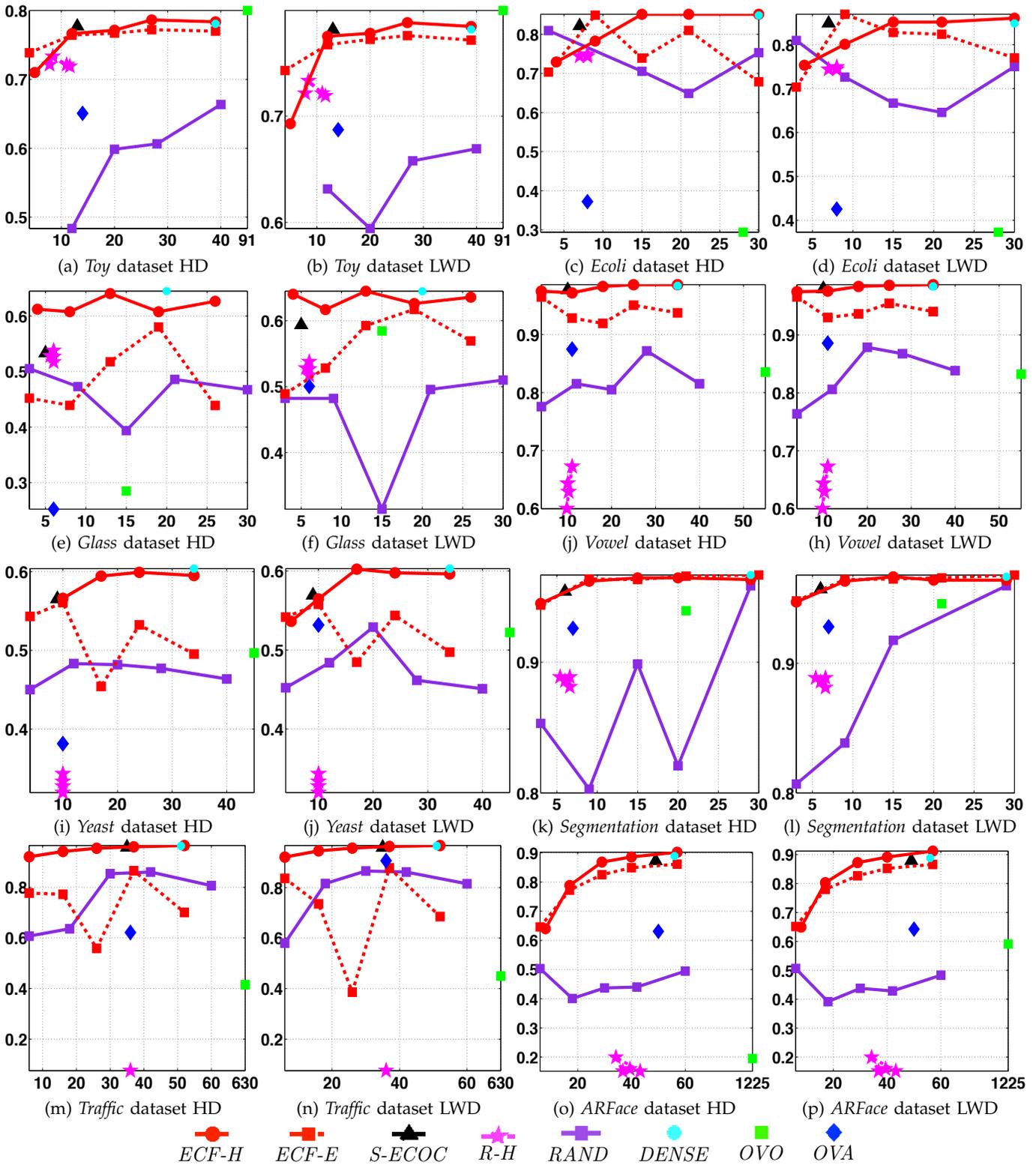


Fig. 9. Multi-class classification accuracy (y axis) as a function of the number of dichotomies for all datasets and both decoding measures (x axis).

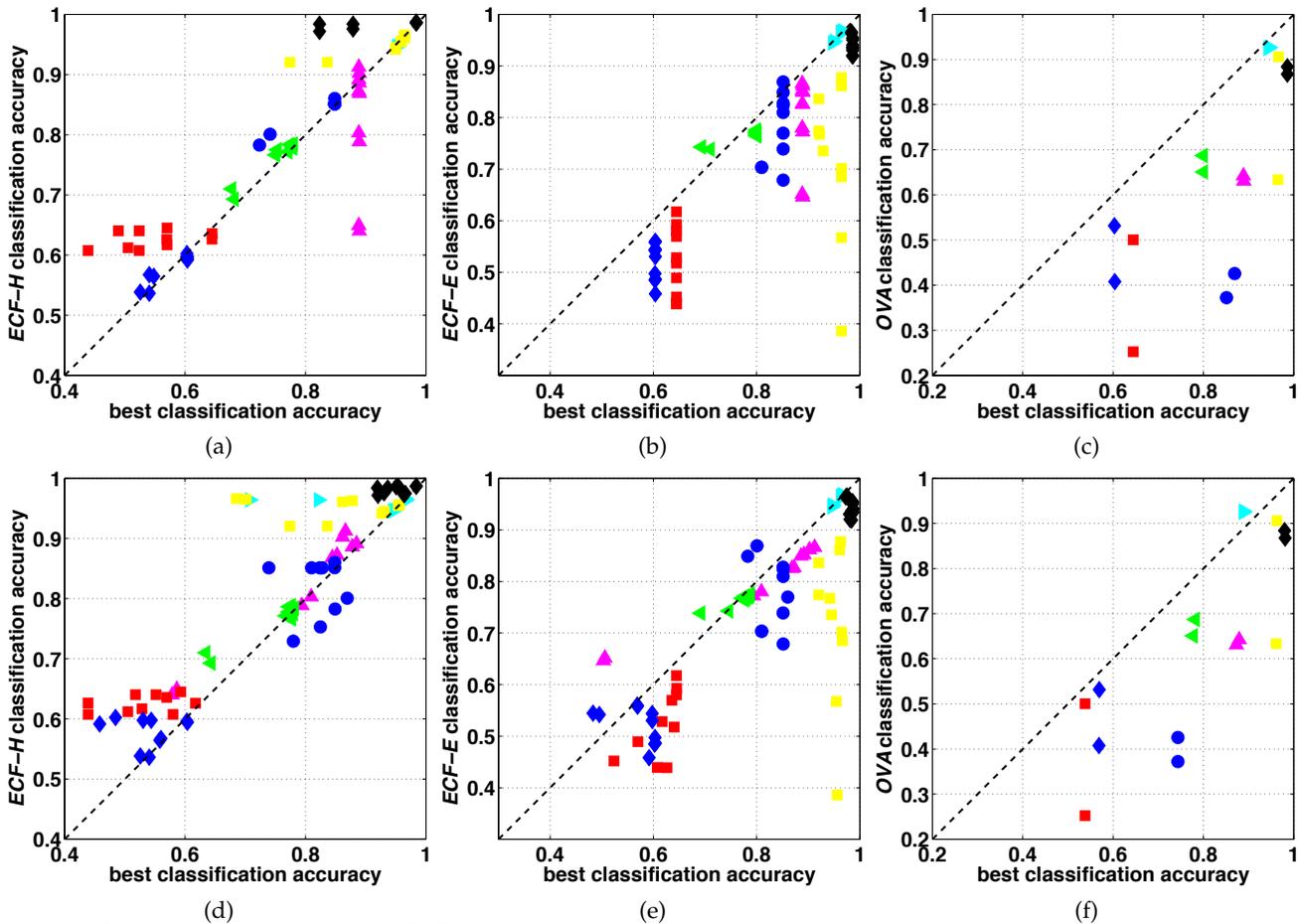


Fig. 10. (a) Summary of performance of *ECF-H* method over all datasets using the number of SVs and the number of dichotomies as the measure of complexity, respectively for *ECF-H* (a)(d), *ECF-E* (b)(e) and *OVA* (c)(f).

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 prioritized 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.

REFERENCES

- [1] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David J Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *ICAIIS*, pages 11–18, 2007.
- [2] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Journal of Machine Learning Research*, volume 1, pages 113–141, 2002.
- [3] Erin L. Allwein, Robert E. Schapire, Yoram Singer, and Pack Kaelbling. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.
- [4] Miguel Ángel Bautista, Sergio Escalera, Xavier Bar, and Oriol Pujol. On the design of an ecoc-compliant genetic algorithm. *Pattern Recognition*, 47(2):865 – 884, 2013.
- [5] Miguel Bautista, Oriol Pujol, Xavier Baró, and Sergio Escalera. Introducing the separability matrix for error correcting output codes coding. *MCS*, pages 227–236, 2011.

- [6] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Image classification using random forests and ferns. In *ICCV 2007*, pages 1–8. IEEE, 2007.
- [7] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152. ACM, 1992.
- [8] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2009.
- [9] Stephen Boyd and Lin Xiao. Least-squares covariance matrix adjustment. *SIAM Journal on Matrix Analysis and Applications*, 27(2):532–546, 2005.
- [10] 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.
- [11] Lawrence Cayton and Sanjoy Dasgupta. Robust euclidean embedding. In *ICML*, pages 169–176. ACM, 2006.
- [12] Chih-Chung Chang and Chih-Jen 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>.
- [13] Koby Crammer and Yoram Singer. Improved output coding for classification using continuous relaxation. In *NIPS*, volume 13, page 437. MIT Press, 2001.
- [14] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.
- [15] Fernando De la Torre. A least-squares framework for component analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 34(6):1041–1055, 2012.
- [16] 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.
- [17] 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.
- [18] Sergio Escalera, David MJ Tax, Oriol Pujol, Petia Radeva, and Robert PW Duin. Subclass problem-dependent design for error-correcting output codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1041–1054, 2008.
- [19] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785. IEEE, 2009.
- [20] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *COLT*, pages 23–37. Springer, 1995.
- [21] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.
- [22] Keinosuke Fukunaga and Thomas E Flick. An optimal global nearest neighbor metric. *Pattern Analysis and Machine Intelligence, Transactions on*, (3):314–318, 1984.
- [23] Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2072–2079. IEEE, 2011.
- [24] Tianshi Gao and Daphne Koller. Multiclass boosting with hinge loss based on output coding. In *ICML*, pages 569–576, 2011.
- [25] N. Garcia-Pedrajas and C. Fyfe. Evolving output codes for multiclass problems. *Evolutionary Computation, IEEE Transactions on*, 12(1):93–106, 2008.
- [26] Phil Gill. Numerical linear algebra and optimization. 2007.
- [27] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.
- [28] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, pages 1–8. IEEE, 2008.
- [29] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear gaussseidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000.
- [30] YX Gu, Qing Ren Wang, and Ching Y Suen. Application of a multilayer decision tree in computer recognition of chinese characters. *Pattern Analysis and Machine Intelligence, Transactions on*, (1):83–89, 1983.
- [31] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Nnmf: an optimal gradient method for nonnegative matrix factorization. *Signal Processing, IEEE Transactions on*, 60(6):2882–2898, 2012.
- [32] Nicholas J Higham. Computing the nearest correlation matrix problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343, 2002.
- [33] Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *ACM SIGKDD*, pages 1064–1072. ACM, 2011.
- [34] Hyunsoo Kim and Haesun 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.
- [35] Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.
- [36] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.
- [37] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.
- [38] Ana C. Lorena and André C. P. L. F. Carvalho. Evolutionary design of multiclass support vector machines. *Journal of Intelligent Fuzzy Systems*, 18:445–454, October 2007.
- [39] Jérôme Malick. A dual approach to semidefinite least-squares problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):272–284, 2004.
- [40] Marcin Marszałek and Cordelia Schmid. Constructing category hierarchies for visual recognition. In *ECCV*, pages 479–491. Springer, 2008.
- [41] A. Martinez and R. Benavente. The AR face database. In *Computer Vision Center Technical Report #24*, 1998.
- [42] Pauli Miettinen, Taneli Mielikainen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. *Knowledge and Data Engineering, IEEE Transactions on*, 20(10):1348–1362, 2008.
- [43] Indraneel Mukherjee and Robert E Schapire. A theory of multiclass boosting. *The Journal of Machine Learning Research*, 14(1):437–497, 2013.
- [44] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [45] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, pages 503–510. IEEE, 2011.
- [46] O. Pujol, P. Radeva, and J. Vitrià. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 28, pages 1001–1007, 2006.
- [47] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.
- [48] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- [49] Douglas LT Rohde. Methods for binary multidimensional scaling. *Neural Computation*, 14(5):1195–1232, 2002.
- [50] Mohammad J Saberian and Nuno Vasconcelos. Multiclass boosting: Theory and algorithms. In *NIPS*, pages 2124–2132, 2011.
- [51] Robert E Schapire. Using output codes to boost multiclass learning problems. In *ICML*, volume 97, pages 313–321, 1997.
- [52] T.Hastie and R.Tibshirani. Classification by pairwise grouping. *NIPS*, 26:451–471, 1998.
- [53] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.
- [54] Yair Weiss, Rob Fergus, and Antonio Torralba. Multidimensional spectral hashing. In *ECCV 2012*, pages 340–353. Springer, 2012.
- [55] Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.
- [56] Felix X Yu, Liangliang Cao, Rogerio S Feris, John R Smith, and Shih-Fu Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, pages 771–778. IEEE, 2013.
- [57] Xiao Zhang, Lin Liang, and Heung-Yeung Shum. Spectral error correcting output codes for efficient multiclass recognition. In *ICCV*, pages 1111–1118, Sept 2009.
- [58] Bin Zhao and Eric P Xing. Sparse output coding for large-scale visual recognition. In *CVPR*, pages 3350–3357. IEEE, 2013.

- [59] Guoqiang Zhong and Mohamed Cheriet. Adaptive error-correcting output codes. In *IJCAI*, pages 1932–1938. AAAI Press, 2013.
- [60] Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu. Joint learning of error-correcting output codes and dichotomizers from data. *Neural Computing and Applications*, 21(4):715–724, 2012.
- [61] Jin Deng Zhou, Xiao Dan Wang, Hong Jian Zhou, Jie Ming Zhang, and Ning Jia. Decoding design based on posterior probabilities in ternary error-correcting output codes. *Pattern Recognition*, 45(4):1802 – 1818, 2012.
- [62] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.



Sergio Escalera Sergio Escalera received the B.S. and M.S. degrees from the Universitat Autnoma de Barcelona (UAB), Barcelona, Spain, in 2003 and 2005, respectively. He obtained the P.h.D. degree on Multi-class visual categorization systems at Computer Vision Center, UAB. He obtained the 2008 best Thesis award on Computer Science at Universitat Autnoma de Barcelona. He lead the Human Pose Recovery and Behavior Analysis Group at University of Barcelona. His research interests include, between others, machine learning, statistical pattern recognition, visual object recognition, and human computer interaction systems, with special interest in human pose recovery and behavior analysis.



Miguel Ángel Bautista received his B. Sc. and M. Sc. degrees in Computer Science and Artificial Intelligence from Universitat de Barcelona and Universitat Politcnica de Catalunya respectively in 2010. He is a research member at Computer Vision Center at Universitat Autnoma de Barcelona, Applied Math and Analysis Dept. at Universitat de Barcelona and BCN Perceptual Computing Lab and Human Pose Recovery and Behavior Analysis Group at University of Barcelona . In 2010 Miguel Angel received the

first prize from the Catalan Association of Artificial Intelligence Thesis Awards. Currently Miguel Angel is pursuing a Ph. D in Error Correcting Output Codes as a theoretical framework to treat multi-class and multi-label problems. His interests are, between others, Machine Learning, Computer Vision, Convex Optimization and its applications into Human Gesture analysis.



Oriol Pujol Oriol Pujol Vila obtained the degree in Telecommunications Engineering in 1998 from the Universitat Politcnica de Catalunya (UPC). The same year, he joined the Computer Vision Center and the Computer Science Department at Universitat Autnoma de Barcelona (UAB). In 2004 he received the Ph.D. in Computer Science at the UAB on work in deformable models, fusion of supervised and unsupervised learning and intravascular ultrasound image analysis. In 2005 he joined the Dept. of Matemtica Aplicada i Anlisi

at Universitat de Barcelona where he became associate professor. He is member of the BCN Perceptual Computing Lab. He has been since 2004 an active member in the organization of several activities related to image analysis, computer vision, machine learning and artificial intelligence.



Fernando De la Torre is an Associate Research Professor in the Robotics Institute at Carnegie Mellon University. He received his B.Sc. degree in Telecommunications, as well as his M.Sc. and Ph. D degrees in Electronic Engineering from La Salle School of Engineering at Ramon Llull University, Barcelona, Spain in 1994, 1996, and 2002, respectively. His research interests are in the fields of computer vision and Machine Learning. Currently, he is directing the Component Analysis Laboratory (<http://ca.cs.cmu.edu>) and

the Human Sensing Laboratory (<http://humansensing.cs.cmu.edu>) at Carnegie Mellon University. He has over 130 publications in referred journals and conferences. He has organized and co-organized several workshops and has given tutorials at international conferences on the use and extensions of Component Analysis.