Multiclass from Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches

Anderson Rocha, *Member, IEEE*, and Siome Klein Goldenstein, *Senior Member, IEEE*

***Abstract*— Recently, there has been a lot of success in the development of effective binary classifiers. Although many statistical classification techniques have natural multiclass extensions, some, such as the support vector machines, do not. The existing techniques for mapping multiclass problems onto a set of simpler binary classification problems run into serious efficiency problems when there are hundreds or even thousands of classes, and these are the scenarios where this paper's contributions shine. We introduce the concept of correlation and joint probability of base binary learners. We learn these properties during the training stage, group the binary leaner's based on their independence and, with a Bayesian approach, combine the results to predict the class of a new instance. Finally, we also discuss two additional strategies: one to reduce the number of required base learners in the multiclass classification, and another to find new base learners that might best complement the existing set. We use these two new procedures iteratively to complement the initial solution and improve the overall performance. This paper has two goals: finding the most discriminative binary classifiers to solve a multiclass problem and keeping up the efficiency, i.e., small number of base learners. We validate and compare the method with a diverse set of methods of the literature in several public available datasets that range from small (10 to 26 classes) to large multiclass problems (1000 classes) always using simple reproducible scenarios.**

***Index Terms*— Error correcting output codes (ECOC), multiclass from binary, one-versus-all (OVA), one-versus-one (OVO).**

I. Introduction

SUPERVISED learning is a machine learning strategy to create a prediction function from training data. The task of the supervised learner is to predict the value of the function for any valid input object after having a number of domain-related training examples [1]. Many supervised learning techniques aim at binary classification [2]. Many real-world recognition and classification problems, however, often require that we map inputs to one out of hundreds or thousands of possible categories. Often, this is the case on several contemporary vision and pattern recognition problems such as face recognition [3]–[5], fingerprinting identification [6], [7], image categorization [8]–[12], and DNA sequencing [13].

Many researchers have proposed an effective approaches for binary classification in the last years, such as margin and linear classifiers, decision trees, and ensembles [1]. Some of these techniques have natural extensions to multiclass problems (e.g., decision trees), but other powerful and popular classifiers, such as support vector machines (SVMs) [14], are not so easily extensible to a multiclass scenario.

According to [15], how one should approach a multiclass recognition is still an open issue. Should it be performed as a series of binary classifications, or by detection, where a search is performed for each of the possible classes? What happens when some classes are ill sampled, not sampled at all or undefined? In spite of its unsolved nature, it is common in the literature to reduce the multiclass problem to multiple simpler binary classification problems—a process called class binarization. This process, however, has some shortcomings as we shall discuss.

We can see the class binarization process as a mapping of a multiclass problem onto several two-class problems (divide-and-conquer) and the subsequent combination of their outcomes to derive the multiclass prediction [16]. We refer to the binary classifiers used in the process as base learners.

In the literature, there are many possible approaches to reduce multiclass to binary classification problems or conversely to combine binary classifiers toward a multiclass problem. We can safely classify such approaches into three broad groups [17]: 1) one-versus-all (OVA); 2) one-versus-one (OVO); and 3) error correcting output codes (ECOC). In addition, the approaches for multiclass from binary based on ECOC codes usually focus on designing a good ECOC matrix responsible to point out, which classes need to be separated, finding a good base learners for them and combining the base learners in the end in a process called decoding strategy.

Normally a class binarization process involves combining several binary base learners and this process can be highly computational intensive. Therefore, in this paper, our goal is to devise a method, which can find the most representative set of base learners for a given problem yielding high classification rates while being the most efficient possible during training and testing. We focus on the creation of the ECOC matrix responsible to point out, which classes are used in different binary base learners, and on the decoding strategy responsible to combine the outcomes of different base learners toward a final multiclass prediction outcome. As we explain later, for the creation of the ECOC matrix, it is important to choose a feasible number of binary base learners to use (for an efficient learning process). In general, the more base learners we use, the more complex is the overall classification procedure. For the decoding strategy, on the other hand, it is essential to choose a deterministic strategy, robust to ties and errors for classification effectiveness.

Manuscript received June 8, 2012; accepted July 20, 2013. Date of publication August 6, 2013; date of current version January 10, 2014. This work was supported in part by São Paulo Research Foundation - FAPESP under Grant 2010/05647-4, the National Counsel of Technological and Scientific Development - CNPq under Grant 307018/2010-5 and Grant 304352/2012-8, and Microsoft.

The authors are with the Institute of Computing, University of Campinas, São Paulo 13083-970, Brazil (e-mail: anderson.rocha@ic.unicamp.br; siome@ic.unicamp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2274735

When tackling complex multiclass problems, it is not hard to find out that choosing the right features is just the first step toward the solution. With the approach introduced in this paper, we show how to use a small number of simple and fast base learners to achieve good results, no matter the choice of the features capturing the phenomena of interest in a classification problem. This is a relevant issue for large-scale classification problems specially in computer vision [3]–[12].

Here, we introduce an efficient and effective approach to combine binary classifiers to perform large multiclass classification. We present a new treatment for the decoding strategy based on the conditional probabilities in groups of correlated binary classifiers. We introduce the concept of classification outcome correlations among binary classifiers during training and present a principled way to find groups of correlated base learners. We also discuss two additional strategies: one to reduce the number of required base learners in the multiclass classification and another to find new base learners. These two new procedures are optional and can iteratively complement the initial method to improve the overall multiclass classification performance and/or accuracy. We validate the proposed method using datasets from the University of California Irvine Datasets (UCI) repository, National Institute of Standards and Technology Dataset (NIST) digits, and the Amsterdam library of objects (ALOI). We show that our approach provides better (or comparable) classification results than the approaches in the literature.

Finally, a preliminary version of this paper has appeared in [18] but did not consider several of the aspects we discuss here, such as the reduction-improvement iterative optimization, extensive validation, and the mathematical formalization.

We organize this paper in five sections. Section II outlines several multiclass classification methods in the literature. The list of discussed methods is not intended to be complete but to give the reader a flavor of different methods being developed for the problem addressed herein. Section III presents our new Bayesian treatment for combining binary classifiers toward multiclass classification as well as two new strategies to improve the classification performance. Section IV presents the experiments that validate the propositions and compare them with other methods in the literature. Finally, Section V draws a conclusion and points out future research directions.

II. State-of-the-Art

Let $\mathcal{T}$ be a set of base learners that represent a decomposition of a multiclass problem into multiple binary problems, $N_{\mathcal{T}}$ be the size of $\mathcal{T}$ (number of base learners considered), and N_c be the number of classes of a given classification problem.

The literature describes broad groups of approaches for reducing multiclass to binary problems: OVA, OVO, and ECOC methods [16].

1) One-Versus-All: Here, we use $N_{\mathcal{T}} = N_c = O(N_c)$ binary classifiers (base learners) [19], [20]. We train the ith base learner using all patterns of class i as positive $(+1)$ and the remaining class patterns as negative (-1) examples. We classify an input example x to the class with the highest response.
2) One-Versus-One: Here, we use $N_{\mathcal{T}} = \binom{N_c}{2} = O(N_c^2)$ base learners. We train the ijth base learner using all patterns of class i as positive and all patterns of class j as negative examples. In this framework, there are many approaches to combine the outcomes such as voting, and decision directed acyclic graphs [21], [22].
3) Error Correcting Output Codes: Here [23], we use a coding matrix $M \in \{-1, 1\}^{N_c \times N_{\mathcal{T}}}$ to point out, which classes to train as positive and negative examples. In this matrix, each column contains the dichotomies we need to consider to train one base learner (e.g., classes 1 and 2 against classes 3, 4, and 5). Allwein *et al.* [24] have extended this approach and proposed to use a coding matrix $M \in \{-1, 0, 1\}^{N_c \times N_{\mathcal{T}}}$. The jth column of the matrix induces a partition of the classes into two metaclasses. An instance x belonging to a class i is a positive instance when training the jth base learner if and only if $M_{ij} = +1$. If $M_{ij} = 0$, then it indicates that the ith class is not part of the training of the jth base learner. In this framework, there are many approaches to combine the outcomes such as voting, Hamming and Euclidean distances, and loss-based functions [25]. When the base learners are margin-based learners, Allwein *et al.* [24] have showed the advantage and the theoretical bounds of using a loss-based function of the margin. Klautau *et al.* [26] have further extended such bounds to other functions.

Note that the computational complexity $O(N_c)$ here refers to the number of required base learners to perform a classification. It is not the only measure necessary to calculate the time required for training and testing. For training, we need to consider the number of training examples of each base learner, the number of base learners used (i.e., the size of each metaclass), and the complexity of each binary classifier with respect to the number of minimum required operations to perform a classification. For testing, we need to consider the number of base learners used and the complexity of each one.

In the case of OVO, the time complexity refers to the $O(N_c^2)$ necessary base learners, each one requiring positive and negative examples of the two classes being trained each time. For testing, OVO requires $O(N_c^2)$ binary classifications for each tested instance.

In the case of OVA, the time complexity refers to the $O(N_c)$ base learners, each one requiring positive examples of the class of interest and negative examples of all the remaining classes. For testing, OVA requires $O(N_c)$ binary classifications for each tested instance.

Pedrajas *et al.* [16] have proposed to combine the strategies of OVO and OVA. Although the combination improves the overall classification, the proposed approach uses $N_{\mathcal{T}} = \binom{N_c}{2} + N_c = O(N_c^2)$ base learners in the training stage.

Moreira and Mayoraz [27] also developed a combination of different classifiers. They have considered the output of each base learner as a probability of the pattern of belonging to a given class. This method requires $(N_c(N_c + 1))/2 = O(N_c^2)$ base learners. Athisos *et al.* [28] have proposed class embeddings to choose the best base learners.

Pujol *et al*. [17] have presented a heuristic method for learning ECOC matrices representing metaclasses to be used when training base learners based on a hierarchical partition of the class space that maximizes a discriminative criterion. This technique finds the potentially best $N_c - 1 = O(N_c)$ base learners for the classification. Crammer and Singer [29] have proven that the problem of finding optimal discrete codes is NP-complete. Hence, Pujol *et al*. have used a heuristic solution for finding the best candidate metaclasses (dichotomies) to train base learners, but it is still computationally expensive, and the authors only report results for $N_c \leq 28$ classes.

Takenouchi and Ishii [30] have used the information transmission theory to combine ECOC base learners. The authors use the full coding matrix M for the metaclasses, i.e., $N_{\mathcal{T}} = (3^{N_c} - 2^{N_c+1} + 1)/2 = O(3^{N_c})$ entries. The authors only report results for $N_c \leq 7$ classes.

Young *et al*. [31] have used dynamic programming to design an one-class-at-a-time removal sequence planning method for multiclass decomposition. Although their approach only requires $N_{\mathcal{T}} = N_c - 1$ base learners in the testing phase, the removal policy in the training phase is expensive. The removal sequence for a problem with N_c classes is formulated as a multistage decision-making problem and requires $N_c - 2$ classification stages. In the first stage, the method uses N_c base learners. In each one of the $N_c - 3$ remaining stages, the method uses $(N_c(N_c - 1))/2$ base learners. Therefore, the total number of required base learners is $N_c^3 - 4N_c^2 + 5N_c/2 = O(N_c^3)$.

Many researchers have proposed ways of dealing with the unbalancing of OVA approaches to compensate for the use of only $O(N_c)$ binary classifiers [32]–[35]. Still, none of them represents the final answer on the matter because many ECOC-based solutions still require fewer for similar, or even better, classification results.

Passerini *et al*. [2] have introduced a decoding function that combines the margins through an estimation of their class conditional probabilities. The authors have assumed that all base learners are independent and solved the problem of combining their outcomes using a Naïve Bayes approach. Their solution works regardless of the number of selected base learners and can be associated with each one of the aforementioned approaches.

In addition to the general machine learning literature, both recognition and classification are common terms in computer vision as well. From our experience, it is very common to have classification problems in vision spanning hundreds or thousands of classes [8], [9], [12]. A quick search in the literature shows that SVM (a natural two-class classifier) is the most used classifier in these problems showing the importance of the approach we discuss in this paper.

In addition to the efforts in the machine learning community to extend SVMs in a multiclass, we can cite an additional and important work in other fields including computer vision. The efforts in the vision community basically are toward being able to speed up each single binary classifier for a multiclass classification such as [36] who analyzed the complexity of several multiclass SVM-based algorithms and highlighted the computational bottleneck they suffer: comparing the input images with every training image. The authors have proposed an algorithm that overcomes such limitation by voting on class labels based on the k-nearest neighbors (NNs) quickly determined by a vocabulary tree. Vedaldi and Zisserman [37] analyzed efficient additive kernels via explicit feature maps and have proposed approximations that have an indistinguishable performance from the full kernel on a number of standard datasets greatly reducing the training and testing times of SVM classifiers. Tomasi *et al*. [38] have explored the process of learning categories from few examples with multimodel knowledge transfer approaches for adapting SVM classifiers to select and weight appropriately prior knowledge coming from different categories in a multiclass scenario.

Lin *et al*. [39] have explored large-scale image classification problems using fast feature extraction and SVM training. In addition, improving the feature extraction step using a Hadoop scheme[1] that performs feature extraction in parallel using hundreds of mappers, the authors develop a parallel averaging stochastic gradient descent algorithm for training OVA SVM classifiers. The two methods together are able to deal with terabytes of training data. Finally, optimized ECOC codes based on evolutive algorithms have also been considered in [40]. For more references and results in multiclass classification methods, refer to [41] and [42].

III. Multiclass from Binary: Expanding Upon OVO, OVA, and ECOC Traditional Methods

In this section, we present a new treatment for the decoding strategy when combining the outcomes of different base learners toward a multiclass solution. We propose a decoding approach based on the conditional probabilities of groups of correlated base learners. For that, we discuss the concept of learning outcome correlations among binary classifiers and present a principled way to find groups of correlated base learners. The rationale is that it is not always possible to consider the output of all base learners of an input as independent random variables (RVs). Finally, we present one strategy to reduce the number of required base learners in the multiclass classification and another to find new base learners to replace less discriminative ones. We can use these two procedures iteratively to improve the overall multiclass classification performance.

A. Formalization

To classify an input, we use a set of trained base learners $\mathcal{T}$. We call $\mathcal{O}_{\mathcal{T}}$ the set of outcomes or realization of the base learners in $\mathcal{T}$. For simplicity, each element of $\mathcal{T}$ is a base learner that produces an outcome $\in \{-1, +1\}$. Therefore, a realization of $\mathcal{T}$ consists of the outcomes $\in \{-1, +1\}$ for all the chosen binary classifiers over all the training examples. In addition, consider l_i as a class indicator (label) for a problem with N_c classes.

Given an input element x to classify, a realization $\mathcal{O}_{\mathcal{T}}$ contains the information available to determine the class of x and to find $P(y = l_i|x) = P(y = l_i|\mathcal{O}_{\mathcal{T}})$. For instance, for

[1] Available at http://wiki.apache.org/hadoop/FrontPage.

one input example x and a set $\mathcal{T}$ with three base learners, a realization $\mathcal{O}_\mathcal{T}$ of $\mathcal{T}$ could be of the form $\mathcal{O}_\mathcal{T}(x) = \langle +1, -1, -1\rangle$.

From the Bayes theorem

$$P(y = l_i|\mathcal{O}_\mathcal{T}) = \frac{P(\mathcal{O}_\mathcal{T}|y = l_i)P(y = l_i)}{P(\mathcal{O}_\mathcal{T})} \propto P(\mathcal{O}_\mathcal{T}|y = l_i)P(y = l_i). \quad (1)$$

$P(\mathcal{O}_\mathcal{T})$ is common to all classes and can be suppressed.

Previous approaches have solved the above model by considering the independence of the outcomes for the base learners in $\mathcal{T}$ [2]. If we consider independence among all the outcomes of base learners in $\mathcal{T}$ for a test example x, the model in (1) becomes

$$P(y = l_i|\mathcal{O}_\mathcal{T}) \propto \prod_{t \in \mathcal{T}} P(\mathcal{O}^t_\mathcal{T}|y = l_i)P(y = l_i) \quad (2)$$

where $\mathcal{O}^t_\mathcal{T}$ is the outcome of a base learner $t \in \mathcal{T}$. The class of the input x is given by

$$\text{class}(x) = \arg\max_i \prod_{t \in \mathcal{T}} P(\mathcal{O}^t_\mathcal{T}|y = l_i)P(y = l_i). \quad (3)$$

B. Relaxing the Independence Assumption

Although the independence assumption simplifies the model, it comes with the limitations and it is not the best choice in all cases [43]. In fact, it is possible that relaxing such constraint might lead to better classification performance.

We relax the assumption of independence among all binary classifiers. When the outcomes of two base learners overlap up to some point for the same training examples, it would be unwise to treat their results as independent RVs. In our approach, we find groups of correlated base learners and represent their outcomes for training examples as dependent RVs, using a single conditional probability table (CPT) as an underlying distribution model. Each group of correlated classifiers then has its own CPT, and we combine the groups as if they are independent from each other—to avoid a dimensionality explosion.

We can view this technique as a Bayesian-network-inspired approach for RV estimation. We decide the RV that represents the class of an input example based on the RVs that show the outcomes of the base learners. We model the multiclass classification problem conditioned to groups of correlated base learners $\mathcal{C}$. The model in (1) then becomes

$$P(y = l_i|\mathcal{O}_\mathcal{T}, \mathcal{C}) \propto P(\mathcal{O}_\mathcal{T}, \mathcal{C}|y = l_i)P(y = l_i). \quad (4)$$

We assume independence only among the groups of highly correlated base learners $c_i \subset \mathcal{C}$. Therefore, the class of an input x is given by

$$\text{class}(x) = \arg\max_j \prod_{c_i \subset \mathcal{C}} P(\mathcal{O}^{c_i}_\mathcal{T}, c_i|y = l_j)P(y = l_j) \quad (5)$$

where $\mathcal{O}^{c_i}_\mathcal{T}$ is the outcome of all base learners in the group of highly correlated base learners $c_i \subset C$ for training data points X. To find the groups of correlated base learners $\mathcal{C}$, we define a correlation matrix $\mathcal{A}$ among the classifiers. The correlation matrix measures how correlated are two base learners when classifying a set of training examples X. In Section III-E, we show how to create such correlation matrix $\mathcal{A}$. After calculating the correlation matrix $\mathcal{A}$, we use a clustering algorithm to find the groups of correlated base learners in $\mathcal{A}$ (Section III-F).

C. Strategies to Improve Efficiency and/or Effectiveness

The groups of correlated base learners can contain classifiers that do not contribute significantly to the overall classification especially if they are within a large interdependent group. We: 1) identify and remove the less important base learners within a group, speeding up the overall classification process and making more robust CPTs estimations or 2) find new base learners that will improve the classification outcome.

In Section III-G, we show a consistent approach to eliminate the less important base learners within groups of correlated base learners. In addition, in Section III-H, we discuss a simple idea to find new base learners that can be used, among other situations, to replace the ones considered as less discriminatives. We can use both procedures iteratively until a convergence criterion is satisfied. These two procedures are very fast as most of the information needed is already calculated during the earlier training steps.

D. General Algorithm

Algorithm 1 presents the main steps of our approach. Lines 1–19 refer to the training part while Line 20 denotes the testing part. In Line 1, we divide the training data into two parts, one for training the different base learners and one part to validate them and to construct the CPTs representing the groups of classifiers that are correlated. In Lines 2–5, we train and validate each base learner using a selected method (e.g., any binary classifier).

Without loss of generality, each base learner produces an output $\in \{-1, +1\}$ for each input x. In Line 6, $\mathcal{O}_\mathcal{T}$ contains all outcomes of the available base learners for all input samples in X_2. In Lines 7 and 8, we find groups of correlated base learners using the correlation matrix $\mathcal{A}$ calculated upon the base learners outcomes $\mathcal{O}_\mathcal{T}$. From the groups of correlated base learners during training, in Line 9, we create a CPT of each group. These CPTs provide the joint probabilities of the outcomes $\mathcal{O}_\mathcal{T}$ and the groups of correlated classifiers $c_i \subset \mathcal{C}$ when testing an unseen input data x_t.

Lines 10 and 11 prepare the algorithm for further optimization if necessary. If further optimizations are desired, Lines 12–18 perform them. In Line 13, we perform the simplification stage to select the best base learners in the multiclass process and discard the less important ones. In Line 14, we use the classification error found in the training to find new base learners and replace the less important ones previously found. Note that in each iteration there are only a few of them and only the new ones need to be retrained considering the training data X. We repeat Lines 13–17 until a convergence criterion is satisfied. In our case, if the classification training error produced by the updated base learners is bigger than the error

Algorithm 1 Multiclass from Binary Using Groups of Correlated Base Learners

Require: Training data points X in N_d dimensional space, testing data point x_t also in N_d dimensional space, set of base learners $\mathcal{T}$, parameter for further optimization $opt \in \{0, 1\}$, and the kind of base learner $method$ to be used (e.g., SVM, linear discriminant analysis (LDA), etc.).

1: **Split** X into two parts, X_1 for training and X_2 for validation
2: **for each** base learner $b_i \in \mathcal{T}$ **do**
3: $ClassificationModel_i \leftarrow$ TRAIN$(X_1, b_i, method)$;
4: $\mathcal{O}_i \leftarrow$VALIDATE$(X_2, b_i, ClassificationModel_i, method)$;
5: **end for**
6: $\mathcal{O}_\mathcal{T} \leftarrow \bigcup \mathcal{O}^i$;
7: **Create** the correlation matrix $\mathcal{A}$ for $\mathcal{O}_\mathcal{T}$;
8: **Find** the highly correlated groups of base learners, $\mathcal{C}$, from $\mathcal{A}$
9: **Create** a CPT for each group $c \subset \mathcal{C}$ of highly correlated classifiers using $\mathcal{O}$;
10: $k \leftarrow 0$;
11: $(\mathcal{C}^{(k)}, \mathcal{T}^{(k)}) \leftarrow (\mathcal{C}, \mathcal{T})$;
12: **if** $opt = 1$ **then** ▷ Further optimizations desired
13: **while** not convergence or stopping criteria **do**
14: **Perform Simplification**. $(\mathcal{C}^{tmp}, \mathcal{T}^{tmp}) \leftarrow$ SIMPLIFICATION$(\mathcal{C}^{(k)}, \mathcal{T}^{(k)}, X, method)$; ▷ Optimization 1
15: **Perform Replacement**. $(\mathcal{C}^{(k+1)}, \mathcal{T}^{(k+1)}) \leftarrow$ REPLACEMENT$(\mathcal{C}^{tmp}, \mathcal{T}^{tmp}, X, method)$; ▷ Optimization 2
16: $k \leftarrow k + 1$;
17: **end while**
18: **end if**
19: $(\mathcal{C}^{(final)}, \mathcal{T}^{(final)}) \leftarrow (\mathcal{C}^{(k)}, \mathcal{T}^{(k)})$;
20: **Classify** x_t according to 5 using the set of base learners $\mathcal{T}^{final}$ and groups of highly correlated base learners $\mathcal{C}^{final}$.

of the previous iteration. Finally, Line 19 updates the final set of base learners considered as well as their relationship during training and Line 20 denotes the testing for any unseen testing data point x_t.

E. Correlation Matrix $\mathcal{A}$

Given a training dataset X consisting of data points in any N_d dimensional space, we measure the correlation between two base learners trained over such data points by considering their binary outcomes $\in \{-1, +1\}$, $\mathcal{O}_i$, $\mathcal{O}_j$

$$\mathcal{A}_{i,j} = \frac{1}{N}\left|\sum_{\forall x \in X} \mathcal{O}_i(x)\mathcal{O}_j(x)\right| \quad \forall\ \mathcal{O}_i \text{ and } \mathcal{O}_j \in \mathcal{O}_\mathcal{T}. \quad (6)$$

According to this model, if two base learners have the same output for all data points in X, their correlation is one. For instance, this is the case when $\mathcal{O}_i = \mathcal{O}_j$. If $\mathcal{O}_i \neq \mathcal{O}_j$ in all cases, their correlation is also one as they always disagree. On the other hand, if two base learners have half outputs different and half equal, their correlation is 0. Using this model, we can group base learners that produce similar outputs and, eventually, eliminate those, which do not contribute significantly to the overall classification procedure.

F. Defining the Groups of Correlated Base Learners

Given a correlation matrix $\mathcal{A}$ representing the relationships among all base learners in a set $\mathcal{T}$, we want to find the group-dependent classifiers while keeping the groups independent from each other. This is a typical problem of clustering an unknown number of groups.

1) Spectral Clustering: Fisher and Poland [44], have introduced a spectral clustering approach, which we use in this paper. Instead of considering two points similar if they are connected by a high-weight edge, the authors propose to assign them a high weight if the overall graph conductivity between them is high. These considerations exhibit an analogy to electrical networks: the conductivity between two nodes depends not only on the conductivity of the direct path between them, but also on other indirect paths to the nodes.

To find the conductivity for any two points in our problem, we consider the correlation matrix $\mathcal{A}$ as a graph in which the entry A_{ij} is the weight between nodes i and j. The first step is then to calculate the conductivity matrix for all pairs of nodes in $\mathcal{A}$. In particular, the conductivity of any pair of nodes n_p and n_q is given by the system of linear equations

$$G\vec{\varphi} = \vec{\eta} \quad (7)$$

where G is a matrix constructed from the original matrix $\mathcal{A}$

$$G(p,q) = \begin{cases} \text{for } p = 1: & \begin{cases} 1, \text{ for } q = 1 \\ 0, \text{ otherwise} \end{cases} \\ \text{otherwise:} & \begin{cases} \sum_{k \neq p} \mathcal{A}(p,k), & \text{for } p = q \\ -\mathcal{A}(p,q), & \text{otherwise} \end{cases} \end{cases} \quad (8)$$

and $\vec{\eta}$ is a vector representing points for which we compute the conductivity

$$\vec{\eta}(k) = \begin{cases} -1, & \text{for } k = p \text{ and } p > 1 \\ 1, & \text{for } k = q \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Then, the conductivity between n_p and n_q, $p < q$, due to the way $\vec{\eta}$ is constructed, is given by

$$\beta(p,q) = \frac{1}{G^{-1}(p,p) + G^{-1}(q,q) - G^{-1}(p,q) - G^{-1}(q,p)}. \quad (10)$$

Because of symmetry, $\beta(p,q) = \beta(q,p)$ and it is necessary to compute G^{-1} only once. After building the conductivity matrix from $\mathcal{A}$, we find groups using any simple cluster method such as KLines [44].

G. Eliminating Less Important Base Learners: Optimization 1

With this strategy, we find the base learners within a group of correlated base learners that are more relevant for the overall multiclass classification. For that, we find the cumulative entropy of each classifier within a group from the examples in the training data X. The higher the cumulative entropy, the more representative is a specific base learner. Let h_{ij} be the cumulative entropy for the classifier j within a group i. We define h_{ij} as

$$h_{ij} = \sum_{l \in L} \sum_{x \in X} \left(p_l^x \log_2(p_l^x) + (1 - p_l^x) \log_2 (1 - p_l^x) \right) \quad (11)$$

where $p_l^x = P(y = l \mid x, c_{ij}, \mathcal{O}_{c_{ij}}(x))$, c_{ij} is the jth base learner within the group $c_i \subset \mathcal{C}$, $\mathcal{O}_{c_{ij}}(x)$ is its outcome for the input x, and $l \in L$ a class label.

We choose the classifiers with the highest cumulative entropy to select the best classifiers within a group (e.g., top 80%). Note that this is a fast procedure as we already have the required probabilities stored in the computed CPTs.

With this simple strategy, we measure the correlation between the base learners with respect to the training data and learn their relationship pointing out if some base learners are really necessary or worthwhile. Being necessary and worthwhile are two different concepts. Sometimes, a base learner may be in the set because it is critical for discriminating between some particular classes. If so, it is unlikely it will share a group of highly correlated classifiers because it would require this base learner to be highly correlated with all other base learners in such group. We have performed some experiments to test this and, in all tested cases, such base learners specific for rare classes are kept in the final pool of base learners.

H. Replacing Less Important Base Learners: Optimization 2

As shown in Section III-G, we are able to find the base learners that are more relevant for the overall multiclass performance with a simple approach. This leads us to the natural consequence of finding the new base learners to replace the ones that are less discriminative. We use the classification confusion matrix generated in the training stage. From such confusion matrix, we are able to point out the classes that are in more confusion with each other and build a representation of the confusions through hierarchical clustering to determine the ones that need urgent base learners to solve them.

We sort the clusters according to the sum of the number of confusions normalized by the number of edges connecting the elements in that cluster excluding the self-references. For each cluster, we find the cut that maximizes the separability of the cluster and its confusion [45]. We summarize these procedures in Algorithm 2.

To illustrate this strategy, consider a step-by-step example. Let $\mathcal{M}$ be a confusion matrix for a multiclass problem with five classes, as shown in Table I. Here, the diagonal is already set to zero as we want to find the classes that are in confusion and the diagonal represents the correct classifications rather than the misclassifications.

According to the Algorithm 2, in Line 3, we perform the normalization of the confusion matrix $\mathcal{M}$. After the normalization, high values represent low confusion. We shall present the reason of such normalization shortly. In Line 4, we find a hierarchical representation of the confusions using a simple hierarchical clustering algorithm such as Agnes [46]. We show the resulting hierarchy of confusions in Fig. 1, in which we see that classes l_2 and l_3 are merged at the height (normalized confusion) of 0.86. This is consistent with the information in the confusion matrix $\mathcal{M}$ of Table I, where l_2 and l_3 have 48 confusions in both directions. The classes l_2, l_3, and l_4 are also in high confusion as we see in Table I. On the other hand, the classes l_1 and l_5 have a common cluster with low confusion. Finally, the set $\{l_1 \ldots l_5\}$ represents a confusion comprising all the classes.

Algorithm 2 Replacing Less Important Base Learners

Require: The set of base learners calculated $\mathcal{T}$ and not eliminated in the *Optimization 1* step as well as their realizations $\mathcal{O}_{\mathcal{T}}$ for the training data X, the training data X, and the kind of base learner $method$ to be used (e.g., SVM, LDA, etc.).

1: $\mathcal{M} \leftarrow$ the classification confusion matrix associated with $\mathcal{T}$ and $\mathcal{O}_{\mathcal{T}}$ for training data X;
2: **Set the diagonal** elements of $\mathcal{M}$ to zero;
3: $\mathcal{M}_{ij} \leftarrow 1 - \mathcal{M}_{ij} / \sum_{i,j} \mathcal{M}_{ij}$; ▷ Normalizing $\mathcal{M}$
4: $R \leftarrow$ hierarchical clustering of $\mathcal{M}$ using, for instance, the simple Agnes [46] algorithm;
5: **Sort** R according to the sum of the confusions of each group $r_i \subseteq R$ normalized by the number of edges connecting the elements in the group r_i excluding the self-references;
6: **for each** group $r_i \subseteq R$ **do**
7: $\vec{d} \leftarrow$ the cut that maximizes the separability r_i; ▷ $\vec{d}$ represents the metaclasses for which we need to train a new base learner b_d.
8: **Train** a new base learner, b_d using X and metaclasses $\vec{d}$;
9: $\mathcal{T} \leftarrow \mathcal{T} \cup b_d$;
10: $\mathcal{O}_{\mathcal{T}} \leftarrow \mathcal{O}_{\mathcal{T}} \cup \mathcal{O}_{b_d}$;
11: **end for**
12: **Return** $\mathcal{T}$ and $\mathcal{O}_{\mathcal{T}}$.

TABLE I
EXAMPLE OF A CONFUSION MATRIX $\mathcal{M}$ FOR A MULTICLASS PROBLEM WITH $|L| = 5$ CLASSES (l_i DENOTES THE CLASS LABELS)

	l_1	l_2	l_3	l_4	l_5
l_1	**0**	0	22	13	2
l_2	0	**0**	48	26	2
l_3	22	48	**0**	37	31
l_4	13	26	37	**0**	1
l_5	2	2	31	1	**0**

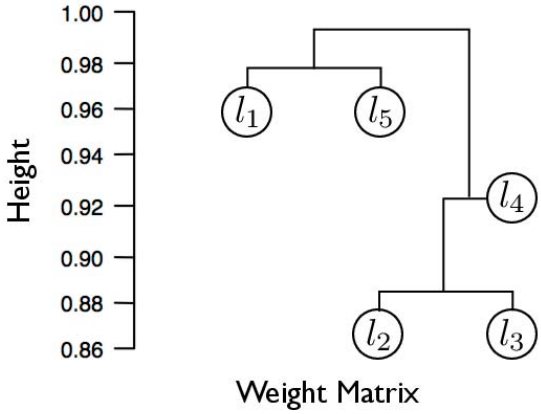

Fig. 1. Hierarchical structure of $\mathcal{M}$.

In Line 5 of the Algorithm 2, we sort the sets according to the sum of the confusions normalized by the number of edges connecting the elements of the set excluding the self-references. For instance, the set in confusion $\{l_2, l_3\}$ has a sum of $96 = 2 \times 48$. As we have two entries excluding the self-references, the normalized weight of this set is 48. On the other hand, the set $\{l_2, l_3, l_4\}$ has a sum of $222 = 2 \times 48 + 2 \times 37 + 2 \times 26$. Given that this sum has six elements excluding the self-references it yields a normalized weight of $37 = 222/6$. In the same way, $\{l_1 \dots l_5\}$ is the third set, in order, given that it has a sum of 364. As this sum has 20 elements excluding the self-references, its normalized weight is $18.2 = 364/20$.

To find the metaclasses representing each set of classes in confusion, we can find a minimum cut in the confusion matrix representing this set of classes. This is why we normalize the confusion matrix $\mathcal{M}$ to have low values representing higher confusions.

Consider the first set $\{l_2, l_3\}$. This is the most trivial case to find the cut given that it has only two elements. Therefore, the metaclasses that represent this set can be $\vec{d}_1 = [0, 1, -1, 0, 0]^T$ or its complement. The set $\{l_2, l_3, l_4\}$, however, has more than two elements, and we need to find a way to partition this set and define its metaclasses. For that, we employ normalized cuts [45] in the submatrix representing this set. In this case, we find the best cut as $\{l_2, l_4\}$ and $\{l_3\}$. We can represent the metaclasses here as $\vec{d}_2 = [0, 1, -1, 1, 0]^T$, designing a base learner specialized in separating elements from the classes l_2, and l_4 from elements of the class l_3 and disregarding the rest.

I. Note on Selecting the Initial Set of Base Learners

Our method depends on the cardinality of the original set of classifiers $\mathcal{T}$ and the configuration of those classifiers (which classes they do consider).

We have performed several experiments on this end and discovered that starting with a random set of base learners, following [24], as coding matrix $M \in \{-1, 0, 1\}^{N_c \times N_\mathcal{T}}$ is very efficient and usually yields good classification results. To ensure that all classes are represented, the process for sampling, which classes go on each side of a metaclass representation for a base learner needs close attention, and we ensure that each class is sampled at least once. As for the number base learners for the initial configuration, the experiments show that we can have good classification results using a few base learners (less than the number of classes in a problem). In the experiments, we explore both the number of initial base learners and also the metapartitions used to create the base learners.

A new practitioner in the field might wonder whether random metapartitions on the classes usually represent good base learners, but there are already several important results in the literature pointing out the power of random codes. Dietterich and Bakiri [47] showed them to perform well in classification problems. In [23], random, exhaustive, hill-climbing search, and other coding methods were used to produce ECOC code matrices for different column lengths. Random codes were also investigated in [48] for combining boosting with ECOC, and it was shown that a random code with a near equal column split of labels was theoretically better. Random codes were also shown to give Bayesian performance if pairs of code words were equidistant [49]. For further reference, Windeatt and Ghaderi [25] discusses several important aspects regarding random selection methods for multiclass learning problems.

There is also an important evidence about the power of random selection methods for feature selection [50], [51], dictionary learning [51], and dimensionality reduction [51], [52].

IV. Experiments and Results

In this section, we compare the proposed methods with: OVO, OVA, and ECOC approaches as well as to the method in [2], who have proposed a Bayesian treatment for decoding assuming independence among all base learners classifiers. We present three different results for our method: one without the two optimization strategies discussed in Section III (M-BAS, for baseline proposed method), one using just the simplification strategy in Section III-G (M-OPT1), and one using both strategies in Section III-H (M-OPT2). We illustrate the behavior of the proposed methods using two different types of base learners: LDA and SVMs. For the case of LDA base learners, we also compare the proposed methods with the multiclass LDA based on Calyampudi Rao's generalization [53] as described in [1] and [54]. In this case, the normal two-class discriminant analysis used in the derivation of the LDA is extended to find a subspace, which appears to contain all of the class variability. For the case of SVM base learners, also compare the proposed methods with four additional methods in the literature: OVO using a bound-constrained formulation (B-SVM), multiclass classification by solving a single optimization problem (SB-SVM), Crammer and Singer's L1 and Crammer and Singer's L2 (Squared Hinge-Loss). We decided to use these four methods because all of them are implemented and fully-optimized in the latest BSVM 2.08 library released on June 2012 [55]. More details about the advantages of each method here can be found in [21], [29], [42], and [56].

We validate the proposed methods using two scenarios, small and large datasets. First, we consider the datasets with a relative small number of classes ($N_c < 30$). For that, we use several UCI,[2] and one NIST[3] datasets. Second, we consider two large scale multiclass applications: one for the Australian Sign Language (Auslan),[4] and one for the ALOI.[5] Table II shows the main properties of the datasets we use in the validation-N_c is the number of classes, N_d the number of features, and N the number of instances.

For the ECOC-based experiments, we selected 10 completely random coding matrices in a process similar to the one presented in [24]. As we explained in Section II, we consider a ternary coding matrix $M \in \{-1, 0, 1\}^{N_c \times N_\mathcal{T}}$. We consider 100 random coding matrices for each possible size of a matrix

[2] Available at http://mlearn.ics.uci.edu/MLRepository.html.
[3] Available at http://yann.lecun.com/exdb/mnist/.
[4] Available at http://mlearn.ics.uci.edu/MLRepository.html.
[5] Available at http://www.science.uva.nl/~aloi/.

TABLE II
SUMMARY OF THE DATASETS USED IN THE VALIDATION

Dataset	Source	N_c	N_d	N
Pendigits	UCI	10	16	10,992
MNist digits	NIST	10	785	10,000
Vowel	UCI	11	10	990
Isolet	UCI	26	617	7,797
Letter-2	UCI	26	16	20,000
Auslan	Auslan	95	128	2,565
ALOI	ALOI	1,000	128	108,000

(number of considered base learners). For each coding matrix, we perform fivefold cross validation. In all experiments, we used both LDA and standard SVMs [1] with radial basis function kernel as the base learners (examples of a week and a strong classifier). For the clustering stage in our solution, we report results using Fisher and Poland technique, as we showed in Section III-F. For all experiments using OVO and OVA, we also perform fivefold cross validation.

For the OVO experiments, we show not only the OVO results when considering all one-by-one combinations of base learners but also what happens if we have to use (e.g., due to efficiency or memory allocation constraints) only a random subset of such base learners instead of the full $O(N_c^2)$ set.

A. Scenario 1 (10–26 Classes)

In Figs. 2–4, we compare the proposed methods (M-*) with ECOC based on Hamming decoding (ECOC), OVO, OVA, Passerini's naïve-Bayesian approach (PASSERINI) [2] and other methods in the literature.

From the experiments, we can see that the use of conditional probabilities and groups of correlated base learners to decode the binary classifications and create a multiclass prediction improves the performance of ECOC-based approaches. This is also true for other UCI datasets not shown here such as *abalone*, *covtype*, and *yeast*. For the LDA base learner, note that the normal multiclass LDA is not as effective as the proposed methods (e.g., Pendigits, MNist, Vowel, and Letter-2). For the SVM base learner and MNist, and vowel datasets, B-SVM, SB-SVM, Crammer-Singer L1, and Crammer-Singer L2 are all worse than M-* methods as well as the OVA reference. Considering the Pendigits and isolet datasets, the performances are comparable. However, SB-SVM, Crammer-Singer L1 and Crammer-Singer L2 are not easy to parallelize contrary to M-* and Passerini methods, which can delegate the training and testing of their base learners to separate threads/cores easily.

For multiclass problems with a small number of classes (e.g., $N_c \leq 26$, weak base learners (e.g., LDA) benefits even more from the proposed methods than strong ones (e.g., SVMs). This important result shows us that when we have a problem with many classes, it may be worth using weak classifiers (e.g., LDA), which often are considerably faster than strong ones (e.g., SVMs). When using all OVO base learners, OVO produces good results. However, this implies in the use of all one-by-one base learners in testing as well.

For the UCI and NIST datasets, the M-* results are, in average, one standard deviation (SD) above Passerini's results when using SVM and, at least, two SDs above Passerini's when using LDA. We have found that Passerini's assumption of independence is not as robust as the proposed methods when the number of base learners and classes increase (Section IV-B). This is also true for the OVO combinations. As the number of classes grows, we observe that these solutions become less discriminative.

In Fig. 2, we see that our proposed method with the first optimization (M-OPT1) performs well for MNIST and Pendigits datasets. For SVM base learners, the method with the two optimizations (M-OPT2) is slightly better than its main competitor Passerini's solution.

The most interesting observation here is the cutoff we can use to obtain the same performance we would get when using all the OVO base learners. In both datasets, we only need 15 base learners to have good results using the proposed methods, while OVO combinations need all possible base learners present.

For the experiments in this section, the average number of iterations for M-OPT1 was 3.5 and the average number of base learners effectively replaced were 10% to 25%.

We can draw similar conclusions from the experiments in Fig. 3 (vowel dataset). In this case, note that the OVA baseline approach is not effective. On the other hand, the proposed method with the two optimizations (M-OPT2) provides a good classification performance as well as its single version with no optimization (M-BAS). Here, with a cutoff of $\cong 20$ base learners we still obtain good classification results.

In Fig. 4, we present the results for isolet and Letter-2 datasets. All the proposed methods present better performance than Passerini, ECOC, OVA, and OVO solutions. Using LDA base learners, M-BAS is, at least, five SDs more effective than Passerini's approach. Using SVM base learners, this difference is about two SDs. In these experiments, we could use a cutoff of 50 base learners and still obtain acceptable classification results.

A two-sample paired Wilcoxon signed-rank test [57], [58] for UCI and NIST datasets considering the proposed methods (M-*) with respect to OVO, standard ECOC and Passerini methods shows that the proposed methods are statistically better than their counterparts most of the times. For instance, Tables III and IV show the paired Wilcoxon signed-rank test for UCI's vowel and isolet datasets considering SVM base learners. For the case of vowel dataset, all M-* are consistently better than their counterparts. For isolet, M-BAS is better than all its counterparts except for Passerini's method. M-OPT2 is, however, consistently better than any other method for this dataset at a 95% confidence interval. In these statistical tests, we analyze the classifiers by pairs for different numbers of base learners. We do not use any kind of p-value correction. In addition, we only show statistical test for classifiers in which we had how to control the number of base learners used in the process.

B. Scenario 2 (95 to 1000 Classes)

Now we consider two large-scale applications: Auslan ($N_c = 95$), and ALOI ($N_c = 1000$) datasets.

In such applications, OVO as well as B-SVM, SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2

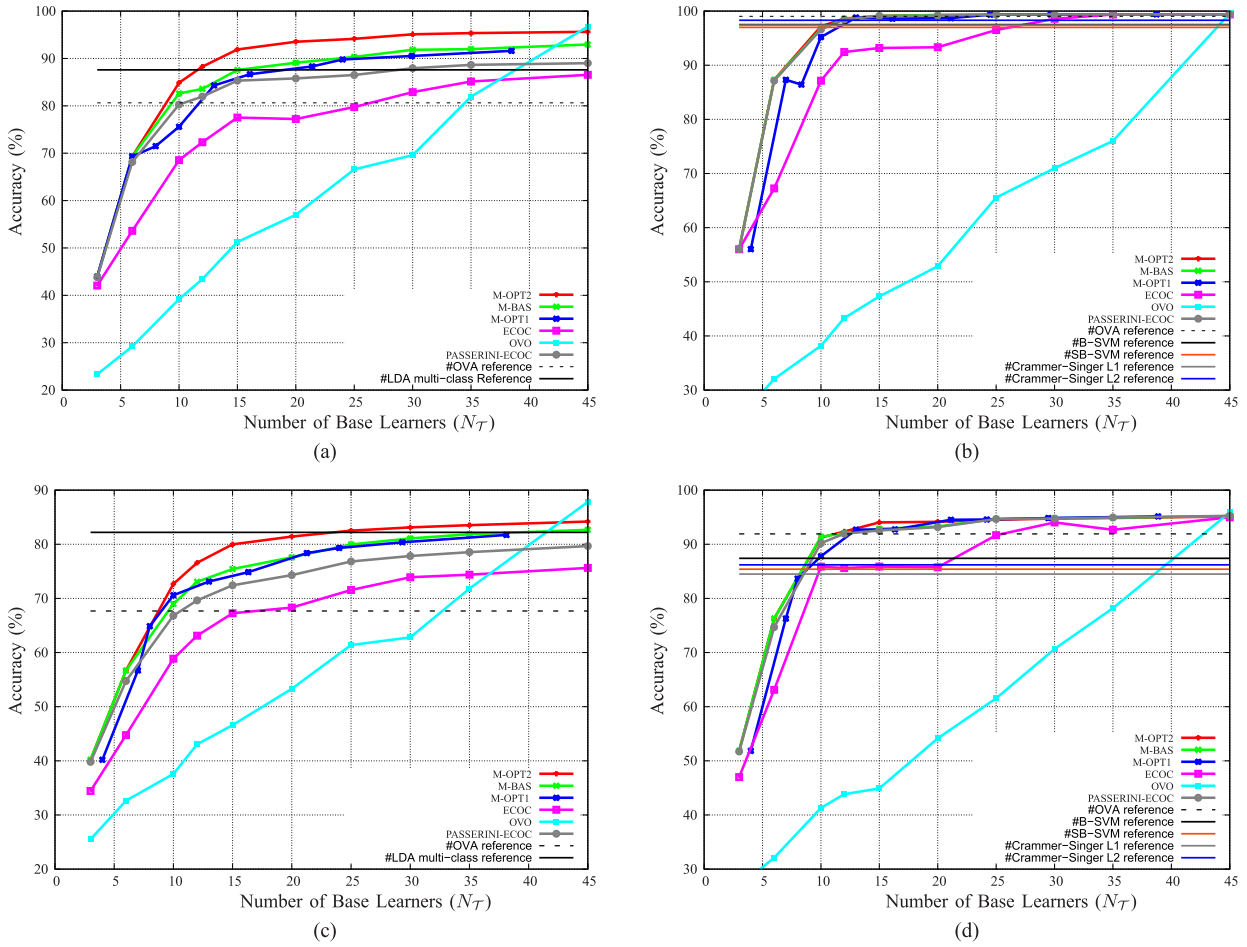

Fig. 2. Proposed methods (M-*) versus ECOC versus OVA versus sampling of OVO versus Passerini for Pendigits, and MNist datasets considering LDA and SVM base learners. For LDA base learner, we also show the multiclass LDA based on Rao's generalization [53]. For the SVM base learner, we also show four methods in the literature: B-SVM, SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2. (a) Pendigits: Base learner = LDA. (b) Pendigits: Base learner = SVM. (c) MNist: Base learner = LDA. (d) MNist: Base learner = SVM.

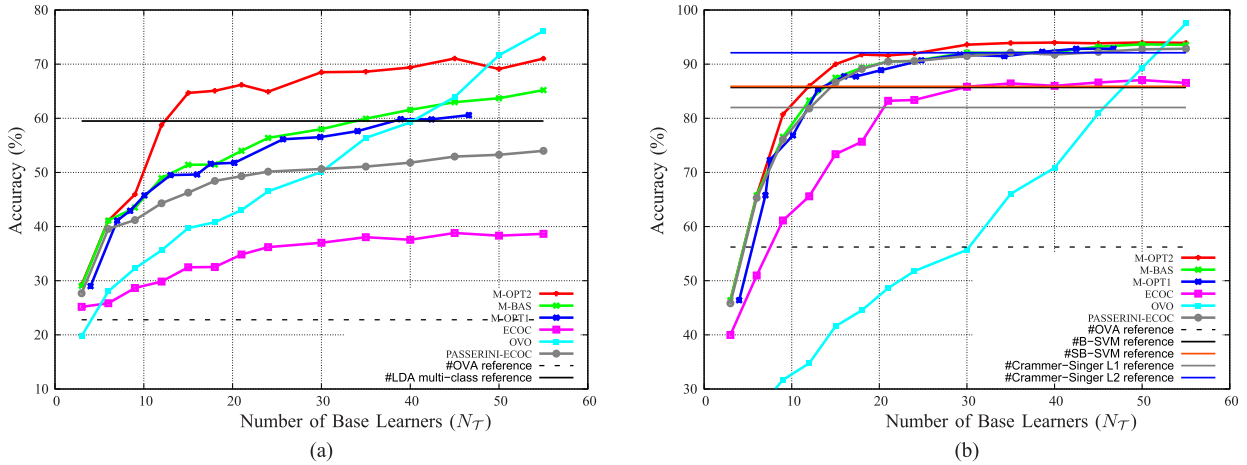

Fig. 3. Proposed methods (M-*) versus ECOC versus OVA versus sampling of OVO versus Passerini for Vowel dataset considering LDA and SVM base learners. For LDA base learner, we also show the multiclass LDA based on Rao's generalization [53]. For the SVM base learner, we also show four methods in the literature: B-SVM, SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2. (a) Vowel: Base learner = LDA. (b) Vowel: Base learner = SVM.

are computationally expensive, and ECOC-based approaches with a few base learners seem to be more appropriate. In Figs. 5–6, we show the results using the baseline method proposed (M-BAS) versus ECOC Hamming decoding and Passerini *et al.* [2] approaches for LDA and SVM base learners. For LDA base learners, we show its multiclass

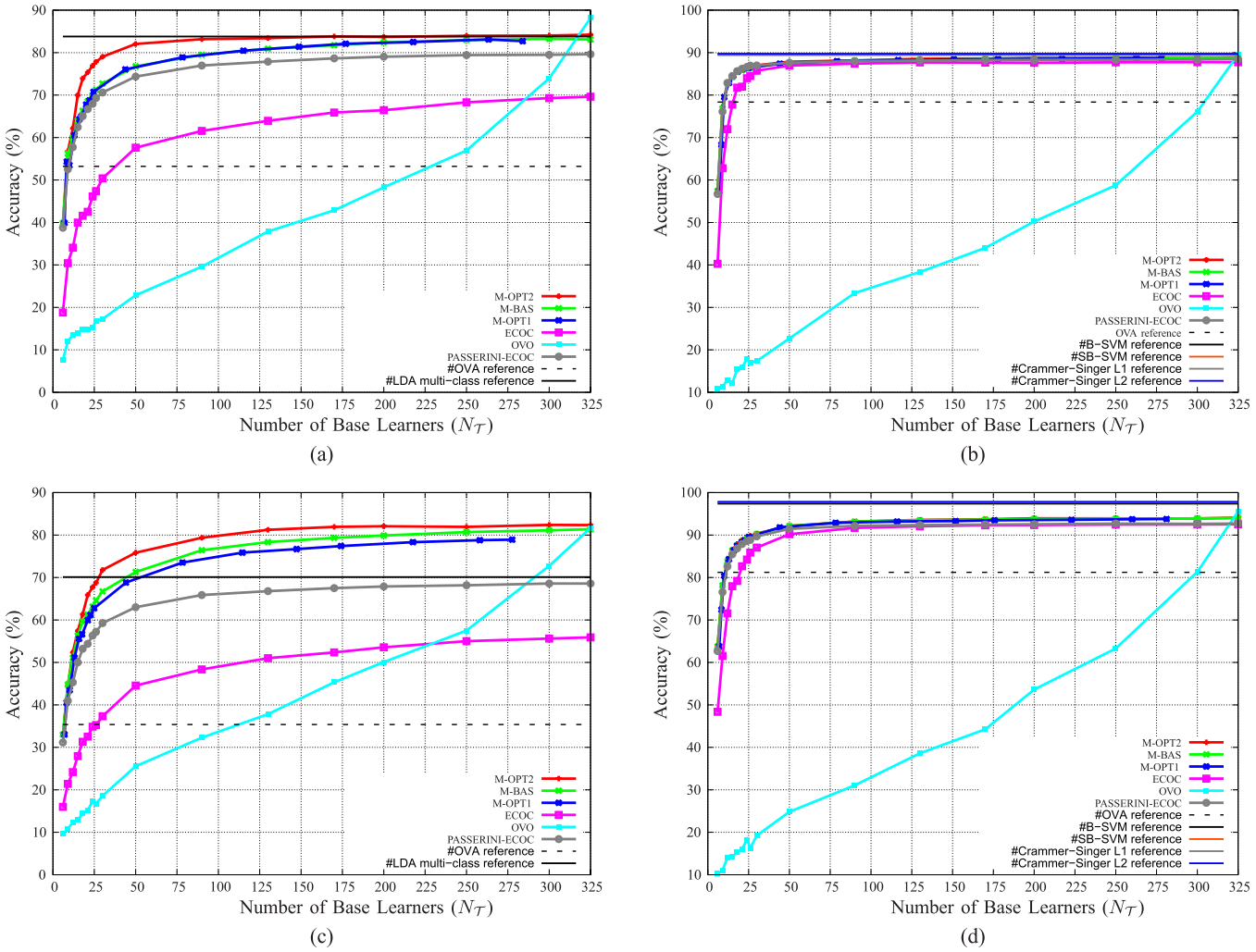

Fig. 4. Proposed methods (M-*) versus ECOC versus OVA versus random sampling of OVO versus Passerini for isolet and Letter-2 datasets considering LDA and SVM base learners. For LDA base learner, we also show the multiclass LDA based on Rao's generalization [53]. For the SVM base learner, we also show four methods in the literature: B-SVM, SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2. (a) Isolet: Base learner = LDA. (b) Isolet: Base learner = SVM. (c) Letter-2: Base learner = LDA. (d) Letter-2: Base learner = SVM.

TABLE III

PAIRED WILCOXON SIGNED-RANK TEST FOR UCI'S VOWEL DATASET CONSIDERING SVM BASE LEARNERS. ALL BLACK ENTRIES REFER TO A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE CONSIDERED METHODS AT A 95% CONFIDENCE INTERVAL. THE LOWER THE p-VALUE THE BETTER. p-VALUES ARE SHOWN UP TO THE FOURTH DECIMAL POINT

	M-BAS	M-OPT1	M-OPT2	ECOC	OVO	Passerini
M-BAS	—	0.003	0.003	0	0	0.002
M-OPT1	0.003	—	0.003	0	0	0.03
M-OPT2	0.003	0.003	—	0	0	0
ECOC	0	0	0	—	0.001	0
OVO	0	0	0	0.001	—	0
Passerini	0.002	0.03	0	0	0	—

TABLE IV

PAIRED WILCOXON SIGNED-RANK TEST FOR UCI'S ISOLET DATASET CONSIDERING SVM BASE LEARNERS. ALL BLACK ENTRIES REFER TO A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE CONSIDERED METHODS AT A 95% CONFIDENCE INTERVAL. THE LOWER THE p-VALUE THE BETTER. p-VALUES ARE SHOWN UP TO THE FOURTH DECIMAL POINT

	M-BAS	M-OPT1	M-OPT2	ECOC	OVO	Passerini
M-BAS	—	0	0.002	0	0	0.378
M-OPT1	0	—	0.001	0	0	0.103
M-OPT2	0.002	0.001	—	0	0	0.023
ECOC	0	0	0	—	0	0
OVO	0	0	0	0	—	0
PASSERINI	0.378	0.103	0.023	0	0	—

version for reference. For SVM, we show B-SVM, SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2 as references.

Here, we emphasize the performance for a small number of base learners in comparison with the number of all possible separation choices. As we increase the number of base learners, all approaches fare steadily better. The experiments show clearly that, in scenarios with more than 30 classes, the independence restriction [2] does not yield the best performance.

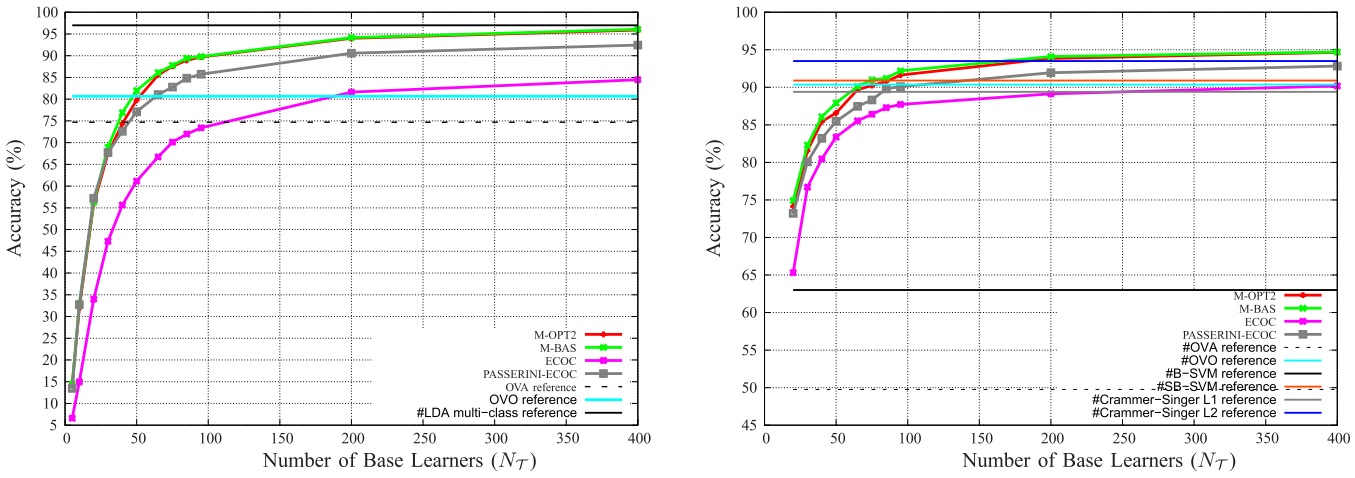

Fig. 5. Proposed methods (M-*) versus ECOC versus OVA versus OVO versus Passerini for Auslan dataset considering LDA and SVM base learners. For LDA base learner, we also show the multiclass LDA based on Rao's generalization [53]. For the SVM base learner, we also show four methods in the literature: B-SVM, SB-SVM, Crammer and Singer's L1 and Crammer and Singer's L2. (a) Auslan: Base learner = LDA. (b) Auslan: Base learner = SVM.

TABLE V
PAIRED WILCOXON SIGNED-RANK TEST FOR AUSLAN DATASET CONSIDERING SVM BASE LEARNERS. ALL BLACK ENTRIES REFER TO A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE CONSIDERED METHODS AT A 95% CONFIDENCE INTERVAL. THE LOWER THE p-VALUE THE BETTER. p-VALUES ARE SHOWN UP TO THE FOURTH DECIMAL POINT

	M-BAS	M-OPT1	M-OPT2	ECOC	OVO	Passerini
M-BAS	—	0.004	0.002	0.002	0.002	0.002
M-OPT1	0.004	—	0.184	0.002	0.002	0.002
M-OPT2	0.002	0.184	—	0.002	0.002	0.002
ECOC	0.002	0.002	0.002	—	0.002	0.002
OVO	0.002	0.002	0.002	0.002	—	0.002
Passerini	0.002	0.002	0.002	0.002	0.002	—

As the image descriptor is not of interest for this paper, for the ALOI dataset we have used an extended color histogram with 128 dimensions [59].

1) Auslan: We show the experiments for Auslan dataset in Fig. 5. In this case, we provide the results for OVO and OVA as baselines. Auslan dataset has $N_c = 95$ classes. Therefore, OVO approach uses $\binom{95}{2} = 4465$ base learners.

OVO with 4465 LDA base learners provides $\cong 80\%$ of classification accuracy while for 95 LDA base learners, The approach in [2] results in $\cong 86\%$, and M-BAS solutions provide $\cong 90\%$ accuracy. As the maximum SD across the cross-validation folds and different executions is $\cong 1\%$, M-BAS is four SDs more reliable than Passerini's solution reducing the classification error in $\cong 30\%$.

With only 400 LDA base learners (approximately 10% of all the OVO possible combinations), M-BAS yields $\cong 96\%$ classification accuracy while Passerini *et al.*'s technique results in $\cong 92.4\%$ accuracy under the same conditions. The maximum SD here is $\cong 0.85\%$. Interestingly, the LDA multiclass fares remarkably well in this dataset. All other methods, including our M-* methods need, at least, 350 base learners to yield a similar classification accuracy. We believe this is a factor regarding the small number of examples for training (about 15–20 per class).

With SVM base learners, OVO provides $\cong 90.3\%$ accuracy. With 95 SVM base learners, Passerini's solution provides $\cong 90\%$ accuracy while M-BAS yields in $\cong 92.2\%$ classification accuracy. The maximum SD here is 0.98%. Using 95 SVM base learners, M-BAS is $\cong 2$ SDs above Passerini and OVO methods.

With 400 SVM base learners (approximately 10% of all the OVO possible combinations), M-BAS yields $\cong 95\%$ classification accuracy or, at least, $\cong 2$ SDs above Passerini *et al.*'s solution and, at least, $\cong 5$ SDs above OVO and the other solutions. Just for the sake of comparison, k-NN[6] (k-NN, $k = 1$) provides $\cong 77\%$ classification accuracy on this dataset. Finally, for this dataset, M-OPT2 is not statistically different than the normal proposed method M-BAS regardless the base learner considered. Note that the only method in the literature comparable with M-* methods is the Crammer-Singer L2. All the other methods fair steadily worse. B-SVM, for instance, leads to a terrible classification result (more than 30% points worse than M-BAS).

Table V shows the paired Wilcoxon signed-rank test for the Auslan dataset considering SVM base learners. Note that all M-* methods are consistently different than its counterparts.

2) Amsterdam Library of Objects: Here, we show the experiment results for the ALOI dataset, which has $N_c = 1000$ classes. In this case, it is not viable to calculate the entire OVO classifications as it would require $\binom{1000}{2} = 499\,500$ base learners in the pool of possible combinations for training and testing. Fig. 6 shows the results for ALOI collection. M-BAS with 200 LDA base learners yields an average classification accuracy of 80% against 68% accuracy of the solution proposed in [2]. Here, the multiclass LDA performs 15% points below the M-BAS methods.

[6]Not shown in the plots.

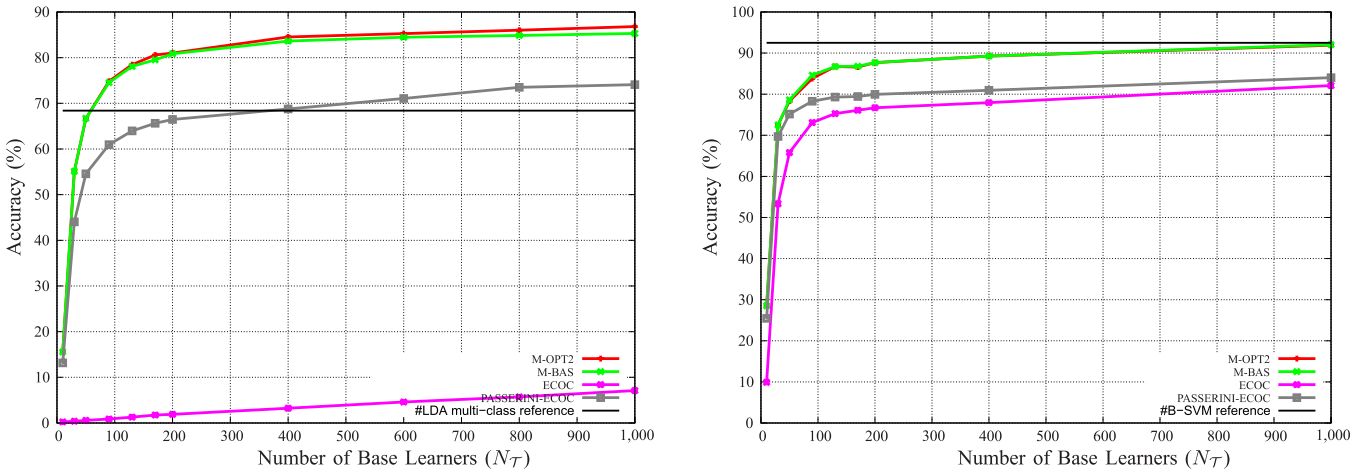

Fig. 6. Proposed methods (M-*) versus ECOC Versus Passerini for ALOI dataset considering LDA and SVM base learners. For LDA base learner, we also show the multiclass LDA based on Rao's generalization [53]. For the SVM base learner, we also show one method from the literature as reference: the B-SVM. The other three approaches: SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2 did produced any result or had memory allocation problems with this dataset. (a) ALOI: Base learner = LDA. (b) ALOI: Base learner = SVM.

TABLE VI
PAIRED WILCOXON SIGNED-RANK TEST FOR ALOI DATASET CONSIDERING SVM BASE LEARNERS. ALL BLACK ENTRIES REFER TO A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE CONSIDERED METHODS AT A 95% CONFIDENCE INTERVAL. THE LOWER THE p-VALUE THE BETTER. p-VALUES ARE SHOWN UP TO THE FOURTH DECIMAL POINT

	M-BAS	M-OPT2	ECOC	Passerini
M-BAS	—	0.106	0.004	0.004
M-OPT2	0.106	—	0.004	0.004
ECOC	0.004	0.004	—	0.004
Passerini	0.004	0.004	0.004	—

TABLE VII
PAIRED WILCOXON SIGNED-RANK TEST FOR ALOI DATASET CONSIDERING LDA BASE LEARNERS. ALL BLACK ENTRIES REFER TO A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE CONSIDERED METHODS AT A 95% CONFIDENCE INTERVAL. THE LOWER THE p-VALUE THE BETTER. p-VALUES ARE SHOWN UP TO THE FOURTH DECIMAL POINT

	M-BAS	M-OPT2	ECOC	Passerini
M-BAS	—	0.013	0.001	0.001
M-OPT2	0.013	—	0.001	0.001
ECOC	0.001	0.001	—	0.001
Passerini	0.001	0.001	0.001	—

In addition, M-BAS with 200 SVM base learners yields $\cong 88\%$ classification accuracy against $\cong 80\%$ of Passerini's method. The maximum SD across the cross-validation folds and different executions here is $\cong 1.2\%$. M-BAS using 1000 SVM base learners has $\cong 93\%$ classification accuracy against Passerini's $\cong 84\%$ and the same as B-SVM. We emphasize that the methods SB-SVM, Crammer and Singer's L1, and Crammer and Singer's L2 had problems when dealing with this more data-intensive dataset. The methods either not converge (produced any result) or had memory allocation problems in the tests.

Tables VI and VII show the paired Wilcoxon signed-rank test for the ALOI dataset considering SVM and LDA base learners, respectively. Note that all M-* methods are consistently different than ECOC and Passerini's solutions. We do not consider OVO here as it is too expensive to computer all 499 500 base learners it requires.

V. CONCLUSION

In recent years, computer vision is finally tackling the recognition and categorization problems, and the desired tasks and applications can now have thousands of classes. In these scenarios, existing methods for binarization of multiclass problems are either impractical (such as OVO) or do not fare so well (such as OVA).

In this paper, we addressed two key issues of multiclass classification: the choice of the coding matrix and the decoding strategy for using when reducing the multiclass problems to binary problems. For that, we presented a new treatment for the decoding strategy.

We have introduced the concept of correlated, or nonindependent, binary classifiers, presented a principled way to group them and use this information to the final inference. We also introduced two optimization approaches: one to reduce the number of required base learners in the multiclass process and eliminate the less discriminative ones and another to automatically find new good base learners. We showed that we can use the two optimization stages iteratively to complement the baseline method and improve the overall multiclass classification performance.

The advantages of the solution we present in this paper are: 1) it works independent of the number of selected base learners; 2) it can be associated with each one of the previous approaches in the literature such as OVO, OVA, ECOC, and

their combinations; 3) it does not rely on the independence restriction among all base learners; 4) its implementation is simple and it uses only basic probability theory; 5) it is fast and does not impact the multiclass procedure; and 6) it is highly parallelizable being suitable to be implemented in current general public utilities (GPUs) or multicore processors.[7]

Unsurprisingly, we have observed that the more saturated the dataset is (when the classification results are close to perfection) the less interesting is the use of the optimization methods (e.g., Pendigits dataset). In this case, the results of our methods and Passerini's method (the most related to ours) are similar. In addition, modeling the conditional tables with a naïve independence assumption tends to work well with a reduced number of classes but might be too restricted for situations that have a large number of classes (e.g., isolet-26, Auslan-200, Aloi-1000). In our experience, the more classes we have and the more confuse is the classification, the better is the scenario for the methods of this paper (M-BAS, M-OPT1, M-OPT2).

Finally, it is wise to take such conclusions carefully. As we all know from the free lunch theorem [60], in machine learning there is no silver bullet to solve all problems. It would be presumptuous to assume the proposed methods can simultaneously deal with any arbitrary small or large number of classes and perform better in all cases. Instead, we prefer to introduce these methods and to claim they are one option for researchers and practitioners, particularly well suited for using with weaker classifiers in situations with hundreds or thousands of classes. As the experiments show, there are cases in which our methods perform clearly better than others but this does not hold for all cases all time. One thing that is clear is that approaches like the ones we propose are in sync with the rapid development of multicore and graphical processing unit's technologies given their straightforward extension for highly parallel architectures and as such we believe the community might benefit from such methods.

Future work includes the design of alternative ways to store the CPTs other than sparse matrices and hashes and the implementation of the current methods using GPUs and multicore machines. As the methods of this paper start from the choice of dichotomies, we also plan to further investigate using these methods on OVO and OVA sets of classifiers.

REFERENCES

[1] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[2] A. Passerini, M. Pontil, and P. Frasconi, "New results on error correcting output codes of kernel machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 15, no. 1, pp. 45–54, Jan. 2004.

[3] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Describable visual attributes for face verification and image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1962–1977, Oct. 2011.

[4] P. J. Phillips, J. R. Beveridge, B. A. Draper, G. Givens, A. J. O'Toole, D. S. Bolme, J. Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer, "An introduction to the good, the bad, & the ugly face recognition challenge problem," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 7758, Mar. 2011.

[7]The source code and datasets are available in R on http://www.ic.unicamp.br/∼rocha/pub/communications or upon request.

[5] G. Chiachia, A. Falcão, and A. Rocha, "Person-specific face representation for recognition," in *Proc. IJCB*, Oct. 2011, pp. 1–8.

[6] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

[7] A. K. Jain, A. A. Ross, and K. Nandakumar, *Introduction to Biometrics*. New York, NY, USA: Springer-Verlag, 2011, pp. 51–96.

[8] J. Deng, A. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" in *Proc. 11th ECCV*, 2010, pp. 71–84.

[9] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Proc. ECCV*, 2010, pp. 57–69.

[10] L. Afonso, J. P. Papa, A. N. Marana, and A. Rocha, "Automatic visual dictionary generation through optimun-path forest clustering," in *Proc. 19th ICIP*, Sep. 2012, pp. 1897–1900.

[11] R. G. Cinbis, J. Verbeek, and C. Schmid, "Image categorization using Fisher kernels of non-iid image models," in *Proc. IEEE Int. CVPR*, Jun. 2012, pp. 1–8.

[12] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2010, pp. 2297–2304.

[13] H. Stranneheim, M. Käller, T. Allander, B. Andersson, L. Arvestad, and J. Lundeberg, "Classification of DNA sequences using bloom filters," *Bioinformatics*, vol. 26, no. 13, pp. 1595–1600, Jul. 2010.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Mar. 1995.

[15] R. P. Duin and E. Pekalska, "Open issues in pattern recognition," in *Computer Recognition Systems*, M. Kurzynski, E. Puchala, M. Wozniak, and A. Zolnierek, Eds. New York, NY, USA: Springer-Verlag, 2005, pp. 27–42.

[16] N. Pedrajas and D. Boyer, "Improving multi-class pattern recognition by the combination of two strategies," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1001–1006, Jun. 2006.

[17] O. Pujol, P. Radeva, and J. Vitria, "Discriminant ECOC: A heuristic method for application dependent design of ECOC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1007–1012, Jun. 2006.

[18] A. Rocha and S. Goldenstein, "From binary to multi-class: Divide to conquer," in *Proc. Int. Conf. Comput. Vis. Theory Appl. VISAPP*, 2009, pp. 323–330.

[19] P. Clark and R. Boswell, "Rule induction with CN_2: Some improvements," in *Proc. EWSL*, Mar. 1991, pp. 151–163.

[20] R. Anand, K. Mehrotra, C. Mohan, and S. Ranka, "Efficient classification for multi-class problems using modular neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 6, no. 1, pp. 117–124, Jan. 1995.

[21] J. Platt, N. Christiani, and J. Taylor, "Large margin dags for multi-class classification," in *Proc. Int. Conf. NIPS*, 1999, pp. 547–553.

[22] S. Abe, "Analysis of multiclass support vector machines," in *Proc. Int. Conf. CIMCA*, 2003, pp. 385–396.

[23] T. Dietterich and G. Bakiri, "Solving multi-class problems via ECOC," *J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 263–286, Jan. 1996.

[24] E. Allwein, R. Shapire, and Y. Singer, "Reducing multi-class to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, no. 1, pp. 113–141, Jan. 2000.

[25] T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multi-class learning problems," *Inf. Fusion*, vol. 4, no. 1, pp. 11–21, 2003.

[26] A. Klautau, N. Jevtic, and A. Orlitsky, "On nearest-neighbor ECOC with application to all-pairs multiclass support vector machines," *J. Mach. Learn. Res.*, vol. 4, no. 1, pp. 1–15, Jan. 2004.

[27] M. Moreira and E. Mayoraz, "Improved pairwise coupling classification with correcting classifiers," in *Proc. Eur. ICML*, Apr. 1998, pp. 160–171.

[28] V. Athisos, A. Stefan, Q. Yuan, and S. Sclaroff, "ClassMap: Efficient multiclass recognition via embeddings," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

[29] K. Crammer and Y. Singer, "On the learnability and design of output codes for multi-class problems," *J. Mach. Learn. Res.*, vol. 47, nos. 2–3, pp. 201–233, Mar. 2002.

[30] T. Takenouchi and S. Ishii, "Multi-class classification as a decoding problem," in *Proc. IEEE Symp. FOCI*, Apr. 2007, pp. 470–475.

[31] C. Young, C. Yen, Y. Pao, and M. Nagurka, "One-class-at-time removal sequence planning method for multi-class problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 17, no. 6, pp. 1544–1549, Nov. 2006.

[32] J. Manikandan and B. Venkataramani, "Design of a modified one-against-all SVM classifier," in *Proc. IEEE Int. Conf. SMC*, Oct. 2009, pp. 1869–1874.

[33] T. Mota and A. Thome, "One-against-all-based multiclass SVM strategies applied to vehicle plate character recognition," in *Proc. IEEE IJCNN*, Jun. 2009, pp. 2153–2159.
[34] Y. Liu and Y. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proc. IEEE IJCNN*, Aug. 2005, pp. 849–854.
[35] Y. Liu, R. Wang, and Y.-S. Zeng, "An improvement of one-against-one method for multi-class support vector machine," in *Proc. IEEE Int. Conf. Mach. Learn. Cybern.*, Aug. 2007, pp. 2915–2920.
[36] T. Yeh, J. Lee, and T. Darrell, "Scalable classifiers for internet vision tasks," in *Proc. IEEE Int. Conf. CVPRW*, 2008, pp. 1–8.
[37] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2010, pp. 3539–3546.
[38] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2010, pp. 3081–3088.
[39] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2011, pp. 1689–1696.
[40] M. Bautista, X. Baro, O. Pujol, P. Radeva, J. Vitria, and S. Escalera, "Compact evolutive design of error-correcting output codes," in *Proc. Supervised Unsupervised Ensemble Methods Appl. Eur. Conf. Mach. Learn.*, Dec. 2010, pp. 119–128.
[41] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Dzeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognit.*, vol. 45, no. 1, pp. 3084–3104, 2012.
[42] C.-P. Lee and C.-J. Lin, "A study on L2-loss (squared hinge-loss) multiclass SVM," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep. 1, Nov. 2012.
[43] A. Narasimhamrthy, "Theoretical bounds of majority voting performance for a binary classification problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1988–1995, Dec. 2005.
[44] I. Fisher and J. Poland, "New methods for spectral clustering," School Comput. Sci. Eng., Hebrew Univ., Jerusalem, Israel, Tech. Rep. IDSIA-12-04, Jun. 2004.
[45] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[46] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, 1st ed. New York, NY, USA: Wiley, 1990.
[47] T. Dietterich and G. Bakiri, "Error-correcting output codes: A general method for improving multiclass inductive learning programs," in *Proc. Int. Conf. AAAI*, 1991, pp. 572–577.
[48] R. Schapire, "Using output codes to boost multiclass learning problems," in *Proc. ICML*, 1997, pp. 313–321.
[49] G. M. James and T. Hastie, "The error coding method and PICT's," *Comput. Graph. Stat.*, vol. 7, no. 1, pp. 377–387, 1998.
[50] A. Blum, "Random projection, margins, kernels, and feature-selection," in *Proc. Int. Conf. SLSFS*, 2006, pp. 52–68.
[51] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.
[52] S. Dasgupta, "Experiments with random projection," in *Proc. Conf. UAI*, 2000, pp. 143–151.
[53] C. R. Rao, "The utilization of multiple measurements in problems of biological classification," *J. R. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 10, no. 2, pp. 159–203, Feb. 1948.
[54] B. Ripley, *Pattern Recognition and Neural Networks*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[55] C.-W. Hsu and C.-J. Lin. (2012, Jun.). *The BSVM Support Vector Machine Library, Vs. 2.08* [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/bsvm/
[56] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," School Comput. Sci. Eng., Hebrew Univ., Jerusalem, Israel, Tech. Rep. 1, Dec. 2001.
[57] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
[58] S. Siegel, *Non-Parametric Statistics for the Behavioral Sciences*. New York, NY, USA: McGraw-Hil, 1956.
[59] R. Stehling, M. Nascimento, and A. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proc. ACM CIKM*, 2002, pp. 102–109.
[60] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Jan. 1997.

Anderson Rocha (M'05) received the B.Sc. degree in computer science from the Federal University of Lavras, Lavras, Brazil, in 2003, and the M.S. and Ph.D. degrees in computer science from the University of Campinas, Unicamp, Brazil, in 2006 and 2009, respectively.

He is currently an Assistant Professor with the Institute of Computing, Unicamp. His current research interests include digital image and video forensics, pattern analysis, machine learning, and general computer vision.

Dr. de Rezende Rocha is an Elected Member of the IEEE Information Forensics and Security Technical Committee. In 2011, he has been elected an Affiliate Member of the Brazilian Academy of Sciences and a fellow of Microsoft Research Faculty.

Siome Klein Goldenstein (SM'94) received the Ph.D. degree in computer and information science from the University of Pennsylvania, Philadelphia, PA, USA, in 2002, the M.Sc. degree in computer science from Pontifícia Universidade Catolica do Rio de Janeiro, Rio de Janeiro, Brazil, in 1997, and the Electronic Engineering degree from the Federal University of Rio de Janeiro, Rio de Janeiro, in 1995.

He is an Associate Professor with the Institute of Computing, University of Campinas, Unicamp, Brazil. He is an Area Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and *Computer Vision and Image Understanding and Graphical Models*. He has been with the program committee of multiple conferences and workshops. His current research interests include computational forensics, computer vision, computer graphics, and machine learning.