MULTI-CLASS FROM BINARY
Divide to conquer

Anderson Rocha and Siome Goldenstein
Institute of Computing
University of Campinas (Unicamp), Campinas, Brazil

Keywords: Multi-class classification, Error correcting output codes, ECOC, Affine Bayes, Bayesian approach.

Abstract: Several researchers have proposed effective approaches for binary classification in the last years. We can easily extend some of those techniques to multi-class. Notwithstanding, some other powerful classifiers (e.g., SVMs) are hard to extend to multi-class. In such cases, the usual approach is to reduce the multi-class problem complexity into simpler binary classification problems (divide-and-conquer). In this paper, we address the multi-class problem by introducing the concept of affine relations among binary classifiers (dichotomies), and present a principled way to find groups of high correlated base learners. Finally, we devise a strategy to reduce the number of required dichotomies in the overall multi-class process.

1 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 seen a number of training examples (Bishop, 2006). Many supervised learning techniques are conceived for binary classification (Passerini et al., 2004). However, a lot of real-world recognition problems often require that we map inputs to one out of hundreds or thousands of possible categories.

Several researchers have proposed effective approaches for binary classification in the last years. Successful examples of such approaches are margin and linear classifiers, decision trees, and ensembles. We can easily extend some of those techniques to multi-class problems (e.g., decision trees). However, we can not easily extend to multi-class some others powerful and popular classifiers such as SVMs. In such situations, the usual approach is to reduce the multi-class problem complexity into multiple simpler binary classification problems. Binary classifiers are more robust to the curse of dimensionality than multi-class approaches. Hence, it is worth dealing with a larger number of binary problems.

A *class binarization* is a mapping of a multi-class problem onto several two-class problems (divide-and-conquer) and the subsequent combination of their outcomes to derive the multi-class prediction (Pedrajas and Boyer, 2006). We refer to the binary classifiers as *base learners* or *dichotomies*.

There are many possible approaches to reduce multi-class to binary classification problems. We can classify such approaches into three broad groups (Pujol et al., 2006): (1) *One-vs-All* (OVA), (2) *One-vs-One* (OVO), and (3) *Error Correcting Output Codes* (ECOC). Also, the multi-class decomposition into binary problems usually contains three main parts: (1) the ECOC matrix creation; (2) the choice of the base learner; and (3) the decoding strategy.

Our focus here is on the creation of the ECOC matrix and on the decoding strategy. For the creation of the ECOC matrix, it is important to choose a feasible number of dichotomies to use. In general, the more base learners we use, the more complex is the overall procedure. For the decoding strategy, it is essential to choose a deterministic strategy robust to ties and errors in the dichotomies' prediction.

In this paper, we introduce a brand new way to combine binary classifiers to perform large multi-class classification. We present a new Bayesian treatment for the decoding strategy, the *Affine-Bayes Multi-class*. We propose a decoding approach based on the conditional probabilities of groups of high-correlated binary classifiers. For that, we introduce

Rocha A. and Goldenstein S.
MULTI-CLASS FROM BINARY - Divide to conquer.
DOI: 10.5220/0001777803230330
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (VISIGRAPP 2009)*, page
ISBN: 978-989-8111-69-2

the concept of affine relations among binary classifiers and present a principled way to find groups of high correlated dichotomies. Furthermore, we present a strategy to reduce the number of required dichotomies in the multi-class process.

Contemporary Vision and Pattern Recognition problems such as face recognition, fingerprinting identification, image categorization, DNA sequencing among others often have an arbitrarily large number of classes to cope with. Finding the right descriptor is just a first step to solve a problem. Here, we show how to use a small number of simple, fast, and weak or strong base learners to get better results, no matter the choice of the descriptor. This is a relevant issue for large-scale classification problems.

We validate our approach using data sets from the UCI repository, NIST, Corel Photo Gallery, and the Amsterdam Library of Objects. We show that our approach provides better results than OVO, OVA, and ECOC approaches based on other decoding strategies. Furthermore, we also compare our approach to Passerini et al. (Passerini et al., 2004), who proposed a Bayesian treatment for decoding assuming independence among all binary classifiers.

2 STATE-OF-THE-ART

Most of the existing literature addresses one or more of the three main parts of a multi-class decomposition problem: (1) the ECOC matrix creation; (2) the dichotomies choice; and (3) the decoding.

In the following, let $\mathcal{T}$ be the team (set) of used dichotomies $\mathcal{D}$ in a multi-class problem, and $N_{\mathcal{T}}$ be the size of $\mathcal{T}$. Recall that N_c is the number of classes[1].

There are three broad groups for reducing multi-class to binary: *One-vs-All*, *One-vs-One*, and *Error Correcting Output Codes* based methods (Pedrajas and Boyer, 2006).

1. **One-vs-All (OVA).** Here, we use $N_{\mathcal{T}} = N_c = O(N_c)$ binary classifiers (dichotomies) (Clark and Boswell, 1991; Anand et al., 1995). We train the i^{th} classifier using all patterns of class i as positive $(+1)$ examples and the remaining class patterns as negative (-1) examples. We classify an input example x to the class with the highest response.
2. **One-vs-One (OVO).** Here, we use $N_{\mathcal{T}} = \binom{N_c}{2} = O(N_c^2)$ binary classifiers. We train the ij^{th} dichotomy 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 obtained outcomes such as *voting*, and *decision directed acyclic graphs* (DDAGs) (Platt et al., 1999).
3. **Error Correcting Output Codes (ECOC)**. Proposed by Dietterich and Bakiri (Dietterich and Bakiri, 1996), in this approach, 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. Allwein et al. (Allwein et al., 2000) have extended such approach and proposed to use a coding matrix $M \in \{-1, 0, 1\}^{N_c \times N_{\mathcal{T}}}$. In this model, the j^{th} column of the matrix induces a partition of the classes into two meta-classes. An instance x belonging to a class i is a positive instance for the j^{th} dichotomy if and only if $M_{ij} = +1$. If $M_{ij} = 0$, then it indicates that the i^{th} class is not part of the training of the j^{th} dichotomy. In this framework, there are many approaches to combine the obtained outcomes such as *voting*, *Hamming* and *Euclidean distances*, and *loss-based functions* (Windeatt and Ghaderi, 2003).

   When the dichotomies are margin-based learners, Allwein et al. (Allwein et al., 2000) have showed the advantage and the theoretical bounds of using a loss-based function of the margin. Klautau et al. (Klautau et al., 2004) have extended such bounds to other functions.

Pedrajas et al. (Pedrajas and Boyer, 2006) have proposed to combine the strategies of OVO and OVA. Although the combination improves the overall multi-class effectiveness, the proposed approach uses $N_{\mathcal{T}} = \binom{N_c}{2} + N_c = O(N_c^2)$ dichotomies in the training stage. Moreira and Mayoraz (Moreira and Mayoraz, 1998) also developed a combination of different classifiers. They have considered the output of each dichotomy as a probability of the pattern of belonging to a given class. This method requires $\frac{N_c(N_c+1)}{2} = O(N_c^2)$ base learners. Athisos et al. (Athisos et al., 2007) have proposed class embeddings to choose the best dichotomies from a set of trained base learners.

Pujol et al. (Pujol et al., 2006) have presented a heuristic method for learning ECOC matrices based on a hierarchical partition of the class space that maximizes a discriminative criterion. The proposed technique finds the potentially best $N_c - 1 = O(N_c)$ dichotomies to the classification. Crammer and Singer (Crammer and Singer, 2002) 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 dichotomies. Even such solution is computationally expensive, and the authors only report results for $N_c \leq 28$.

[1] In the Appendix, we provide a table of symbols.

Takenouchi and Ishii (Takenouchi and Ishii, 2007) have used the information transmission theory to combine ECOC dichotomies. The authors use the full coding matrix M for the dichotomies, i.e., $N_{\mathcal{T}} = \frac{3^{N_c}-2^{N_c+1}+1}{2} = O(3^{N_c})$ dichotomies. The authors only report results for $N_c \leq 7$ classes.

Young et al. (Young et al., 2006) have used dynamic programming to design an one-class-at-a-time removal sequence planning method for multi-class decomposition. Although their approach only requires $N_{\mathcal{T}} = N_c - 1$ dichotomies 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 multi-stage decision-making problem and requires $N_c - 2$ classification stages. In the first stage, the method uses N_c dichotomies. In each one of the $N_c - 3$ remaining stages, the method uses $\frac{N_c(N_c-1)}{2}$ dichotomies. Therefore, the total number of required base learners are $\frac{N_c^3-4N_c^2+5N_c}{2} = O(N_c^3)$.

Passerini et al. (Passerini et al., 2004) have introduced a decoding function that combines the margins through an estimate of their class conditional probabilities. The authors have assumed that all base learners are independent and solved the problem using a Naïve Bayes approach. Their solution works regardless of the number of selected dichotomies and can be associated with each one of the previous approaches.

3 AFFINE-BAYES MULTI-CLASS

In this section, we present our new Bayesian treatment for the decoding strategy: the *Affine-Bayes Multi-class*. We propose a decoding approach based on the conditional probabilities of groups of affine binary classifiers. For that, we introduce the concept of affine relations among binary classifiers, and present a principled way to find groups of high correlated dichotomies. Finally, we present a strategy to reduce the number of required dichotomies in the multi-class classification.

To classify an input, we use a team of trained base learners $\mathcal{T}$. We call $\mathcal{O}_{\mathcal{T}}$ a realization of $\mathcal{T}$. Each element of $\mathcal{T}$ is a binary classifier (dichotomy) and produces an output $\in \{-1, +1\}$. Given an input element x to classify, a realization $\mathcal{O}_{\mathcal{T}}$ contains the information to determine the class of x. In other words, $P(y = c_i|x) = P(y = c_i|\mathcal{O}_{\mathcal{T}})$.

However, we do not have the probability $P(y = c_i \mid \mathcal{O}_{\mathcal{T}})$. From Bayes theorem,

$$\begin{aligned} P(y &= c_i|\mathcal{O}_{\mathcal{T}}) = \frac{P(\mathcal{O}_{\mathcal{T}}|y = c_i)P(y = c_i)}{P(\mathcal{O}_{\mathcal{T}})} \\ &\propto P(\mathcal{O}_{\mathcal{T}}|y = c_i)P(y = c_i) \end{aligned} \quad (1)$$

$P(\mathcal{O}_{\mathcal{T}})$ is just a normalizing factor and it is suppressed.

Previous approaches have solved the above model by considering the independence of the dichotomies in the team $\mathcal{T}$ (Passerini et al., 2004). If we consider independence among all dichotomies, the model in Equation 1 becomes

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

and the class of the input x is $cl(x) = \arg\max_i \prod_{t \in \mathcal{T}} P(\mathcal{O}_{\mathcal{T}}^t|y = c_i)P(y = c_i)$. Although the independence assumption simplifies the model, it comes with limitations and it is not the best choice in all cases (Narasimhamrthy, 2005). In general, it is quite difficult to solve independence without using smoothing functions to deal with numerical instabilities when the number of terms in the series is too large. In such cases, it is necessary to find a suitable density distribution to describe the data, making the solution more complex.

We relax the assumption of independence among all binary classifiers. When two of these dichotomies have a lot in common, it would be unwise to threat their results as independent random variables (RVs). In our approach, we find groups of affine classifiers (high correlated dichotomies) and represent their outcomes as dependent RVs, using a single *conditional probability table* (CPT) as an underlying distribution model. Each group then has its own CPT, and we combine the groups as if they are independent from each other — to avoid a dimensionality explosion.

Our technique can be interpreted as a Bayesian Network inspired approach for RV estimation. We decide the RV that represent the class based on the RVs that represent the outcomes of the dichotomies.

We model the multi-class classification problem conditioned to groups of affine dichotomies $\mathcal{G}_{\mathcal{D}}$. The model in Equation 1 becomes

$$P(y = c_i|\mathcal{O}_{\mathcal{T}}, \mathcal{G}_{\mathcal{D}}) \propto P(\mathcal{O}_{\mathcal{T}}, \mathcal{G}_{\mathcal{D}}|y = c_i)P(y = c_i). \quad (3)$$

We assume independence only among the groups of affine dichotomies $g_i \in \mathcal{G}_{\mathcal{D}}$. Therefore, the class of an input x is given by

$$cl(x) = \arg\max_j \prod_{g_i \in \mathcal{G}_{\mathcal{D}}} P(\mathcal{O}_{\mathcal{T}}^{g_i}, g_i|y = c_j)P(y = c_j). \quad (4)$$

To find the groups of affine classifiers $\mathcal{G}_{\mathcal{D}}$, we define an affinity matrix $\mathcal{A}$ among the classifiers. The affinity matrix measures how affine are two dichotomies when classifying a set of training examples X. In Section 3.1, we show how to create the affinity matrix $\mathcal{A}$. After calculating the affinity matrix $\mathcal{A}$, we use

a clustering algorithm to find the groups of correlated binary classifiers in $\mathcal{A}$. In Section 3.2, we show how to find the groups of affine dichotomies from an affinity matrix $\mathcal{A}$.

The groups of affine classifiers can contain classifiers that do not contribute significantly to the overall classification. Therefore, we can deploy a procedure to identify the less important dichotomies within an affine group and eliminate them. In this stage, we are able to reduce the number of required dichotomies to perform the multi-class classification and hence speed-up the overall process and make robust CPTs estimations. In Section 3.3, we show a consistent approach to eliminate the less important dichotomies within an affine group.

In Algorithm 1, we present the main steps of our model for multi-class classification. In line 1, we divide the training data into five parts and use four parts to train the dichotomies and one part to validate the trained dichotomies and to construct the conditional probability tables. In lines 3–6, we train and validate each dichotomy using a selected method. The method can be any binary classifier such as LDA, or SVM. Each dichotomy produces an output $\in \{-1, +1\}$ for each input x. In line 8, $\mathcal{O}$ contains all realizations of the available dichotomies for the input data X'. In lines 10 and 11, we find groups of affine dichotomies using the realization $\mathcal{O}_i$. Using the information of groups of affine dichotomies, in line 12, we create a CPT for each affine group. These CPTs provide the joint probabilities of a realization $\mathcal{O}_\mathcal{T}$ and the affine groups $g_i \subset \mathcal{G}_\mathcal{D}$ when testing an unseen input data x. In line 13, our approach finds the best dichotomies within the affine groups. This information can be used in the testing phase to reduce the number of used dichotomies.

3.1 Affinity Matrix $\mathcal{A}$

Given a training data set X, we introduce a metric to find the affinity between two dichotomies realizations $\mathcal{D}_i, \mathcal{D}_j$ whose outputs $\in \{-1, +1\}$

$$\mathcal{A}_{i,j} = \frac{1}{N}\left|\sum_{\forall\, x\, \in\, X} \mathcal{D}_i(x)\mathcal{D}_j(x)\right|, \forall\, \mathcal{D}_i \text{ and } \mathcal{D}_j \in \mathcal{T}. \tag{5}$$

According to the affinity model, if two dichotomies have the same output for all elements in X, their affinity is 1. For instance, this is the case when $\mathcal{D}_i = \mathcal{D}_j$. If $\mathcal{D}_i \neq \mathcal{D}_j$ in all cases, their affinity is also 1. On the other hand, if two dichotomies have half outputs different and half equal, their affinity is 0. Using this model, we can group binary classifiers that produce similar outputs and, further, eliminate those which do not contribute significantly to the overall classification procedure.

Algorithm 1 Affine-Bayes Multi-class.

Require: Training data set X, Testing data X^t, a team of binary classifiers $\mathcal{T}$.
1: **Split** X into k parts, X_i such that $i = 1 \ldots k$;
2: **for each** X_i **do** ▷ Inner k-fold cross-validation.
3: $X' \leftarrow X \setminus X_i$;
4: **for each** dichotomy $d \in \mathcal{T}$ **do**
5: $D_{train} \leftarrow$ TRAIN(X', d, $method$);
6: $\mathcal{O}_d^i \leftarrow$TEST($X_i$, d, $method$, D_{Train});
7: **end for**
8: $\mathcal{O}^i \leftarrow \bigcup(\mathcal{O}_d^i)$;
9: **end for**
10: **Create** the affinity matrix $\mathcal{A}$ for $\bigcup \mathcal{O}^i$;
11: **Perform clustering** on $\mathcal{A}$ to find the affine groups of dichotomies $\mathcal{G}_\mathcal{D}$;
12: **Create** a CPT for each group $g \subset \mathcal{G}_\mathcal{D}$ of affine dichotomies using $\mathcal{O}$;
13: **Perform the shrinking**. $\mathcal{GS}_D \leftarrow$ SHRINK($\mathcal{G}_\mathcal{D}$);
14: **for each** $x \in X^t$ **do**
15: **Perform the classification** of x from the model on Equation 4 either using the set of affine dichotomies $\mathcal{G}_\mathcal{D}$ or the shrinked $\mathcal{GS}_\mathcal{D}$.
16: **end for**

3.2 Clustering

Given an affinity matrix $\mathcal{A}$ representing the relationships among all dichotomies in a team $\mathcal{T}$, we want to find groups of classifiers that have similar affinities. We want to find groups of dependent classifiers while the groups are independent from one another. A good clustering approach is important to provide balanced groups of dichotomies. Such balancing is interesting because it leads to simpler conditional probability tables. In this paper, we use a simple, yet effective, greedy algorithm for finding the dependent groups of dichotomies from the affinity matrix.

In our greedy clustering approach, first we find the dichotomy with the highest affinity sum with respect to all its neighbors (row with highest sum in $\mathcal{A}$). After that, we select the neighbors with affinity greater or equal than a threshold t. Next, we check if each dichotomy in the group is affine to the others and select those satisfying this requirement. This procedure results the first affine group. Afterwards, we remove the selected dichotomies from the main team $\mathcal{T}$ and repeat the process until we analyze all available dichotomies. Throughout experiments, we have found that $t = 0.6$ is a good threshold. We use this value in all experiments in this paper.

3.3 Shrinking

Sometimes, when modeling a problem using conditional probabilities, we have to deal with large conditional probability tables which can lead to over-fitting. One approach to cope with this problem is to suppose independence among all dichotomies which results in the smallest possible CPT. However, as we show in this paper, this approach limits the representative power of the Bayes approach. In the following, we show a clever and alternative approach.

In the shrinking stage, we want to find the dichotomies within a group that are more relevant for the overall multi-class classification. We find the accumulative entropy of each classifier within a group from the examples in the training data X. The higher the accumulative entropy, the more representative is a specific dichotomy. Let h_{ij} be the accumulative entropy for the classifier j within a group of affine dichotomies i. We define h_{ij} as

$$h_{ij} = \sum_{c \in C_L} \sum_{x \in X} (p_{cx} \log_2 (p_{cx} (1 - p_{cx}) \log_2 (1 - p_{cx})) \quad (6)$$

where $p_{cx} = P(y = c \mid x, g_i^j, \mathcal{O}_x^{g_i^j})$, g_i^j is the j^{th} dichotomy within the affine group g_i, $\mathcal{O}_x^{g_i^j}$ is its realization for the input x, and $c \in C_L$ the available class labels.

We choose the classifiers with the highest cumulative entropy to select the best classifiers within an affine group. We have found in the experiments, that selecting 60% of the classifiers is a good tradeoff between multi-class overall effectiveness and efficiency. One could use another cutting criteria, such as the maximum CPT size.

During the training phase, our approach finds the affine groups of binary classifiers and marks the most relevant dichotomies within each group. This information can be used afterwards in the testing phase to reduce the number of required classifiers in the multi-class task.

In summary, with our solution, we measure the affinity on the training data to learn the binary classifiers relationship and decision surface. It is a simple and fast way to estimate the distribution. Sometimes, a dichotomy may be in the *team* because it is critical for discriminating between two particular classes. If so, it is unlikely it will share a group of high-correlated classifiers because it would require this dichotomy to be high-correlated with all dichotomies in such group. We have performed some experiments to test that and, in all tested cases, such dichotomies specific for rare classes are kept in the final pool of dichotomies.

4 EXPERIMENTS AND RESULTS

In this section, we compare our *Affine-Bayes Multi-class* approach to: OVO, OVA, and ECOC approaches based on distances decoding strategies. We also compare our approach to Passerini et al. (Passerini et al., 2004) who have proposed a Bayesian treatment for decoding assuming independence among all binary classifiers.

We validate our approach using two scenarios. In the first scenario, we use data sets with a relative small number of classes ($N_c < 30$). For that, we use two UCI[2], and one NIST[3] data sets. In the second scenario, we have considered two large-scale multi-class applications: one for the Corel Photo Gallery (Corel)[4] data set and one for the Amsterdam Library of Objects (ALOI)[5]. Table 1 presents the main features of each data set we have used in the validation. Recall that, N_c is the number of classes, N_d if the number of features, and N is the number of instances.

Table 1: Data sets' summary.

Data set	Source	$\mathbf{N_c}$	$\mathbf{N_d}$	$\mathbf{N}$
Mnist digits	NIST	10	785	10,000
Vowel	UCI	11	10	990
Isolet	UCI	26	617	7,797
Corel	Corel	200	128	20,000
ALOI	ALOI	1,000	128	108,000

In the ECOC-based experiments, we have selected 15 random coding matrices. For each coding matrix, we perform 5-fold cross validation. For each cross-validation fold, we perform a 5-fold cross validation on the training set to estimate the CPTs. In all experiments, we have used the base learners: Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) (Bishop, 2006).

4.1 Scenario 1 (10–26 Classes)

In Figure 1, we compare *Affine-Bayes* (AB) to ECOC based on Hamming decoding (ECOC), One-vs-One (OVO), and Passerini's approach (PASSERINI) (Passerini et al., 2004). In this experiment, *Affine-Bayes* uses two different coding matrices: AB-ECOC, and AB-OVO.

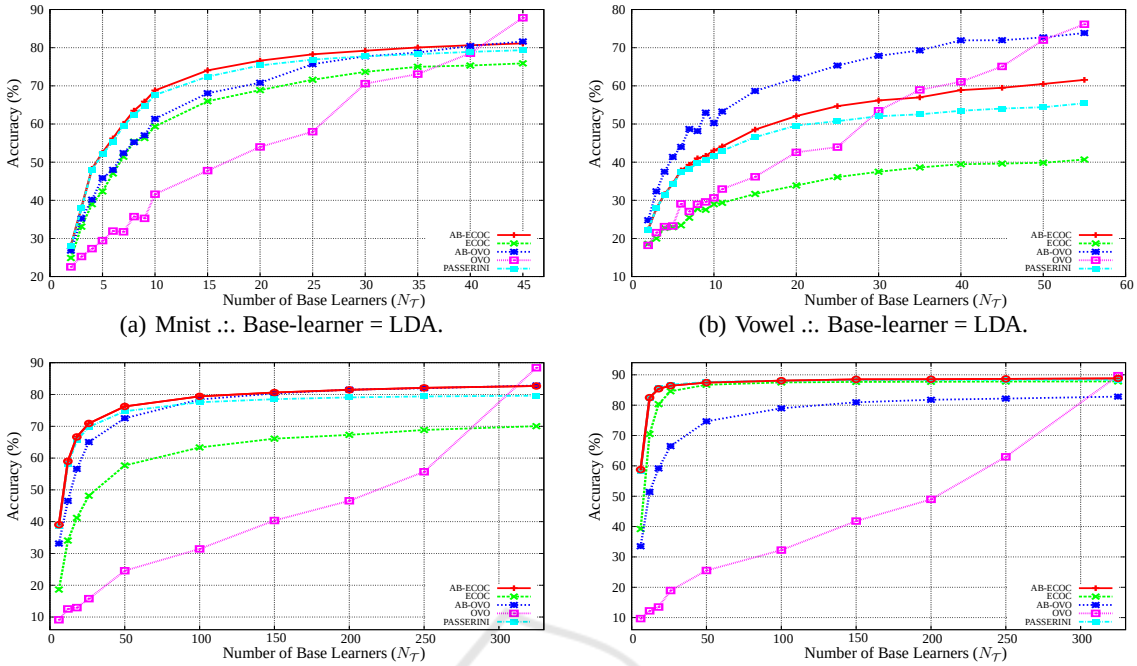
The use of conditional probabilities and affine groups on *Affine-Bayes* to decode the binary classifications and produce a multi-class prediction improves the results for OVO and ECOC coding matrices. This

[2] http://mlearn.ics.uci.edu/MLRepository.html

[3] http://yann.lecun.com/exdb/mnist/

[4] http://www.corel.com

[5] http://www.science.uva.nl/~aloi/



(a) Mnist .:. Base-learner = LDA. (b) Vowel .:. Base-learner = LDA.

(c) Isolet .:. Base-learner = LDA. (d) Isolet .:. Base-learner = SVM.

Figure 1: Affine-Bayes (AB) *vs.* ECOC *vs.* OVO *vs.* Passerini for Mnist, Vowel, and Isolet data sets considering LDA and SVM base learners.

is also true for other UCI data sets not shown here such as *abalone*, *covtype*, *optdigits*, *pendigits*, *vowel*, and *yeast*. Although unsubstantiated here, throughout experiments we have found out that *Affine-Bayes* also improves OVA and ECOC approaches limited to $N_{\mathcal{T}} = N_c$ dichotomies.

Weak classifiers (e.g., LDA) benefits more from *Affine-Bayes* 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 possible, all one-by-one dichotomies (OVO) produce better results. However, random selection of subsets of OVO are not better than ECOC, and *Affine-Bayes* improves both approaches.

For the UCI and Nist small data sets, the *Affine Bayes* results are, in average, one standard deviation above Passerini's results when using SVM and, at least, two standard deviations above when using LDA. However, we have found that Passerini's assumption on independence for all dichotomies is not as robust as *Affine-Bayes* when the number of dichotomies and classes becomes larger (c.f., Sec. 4.2). For small data sets, there is no much gain in using anything sophisticated.

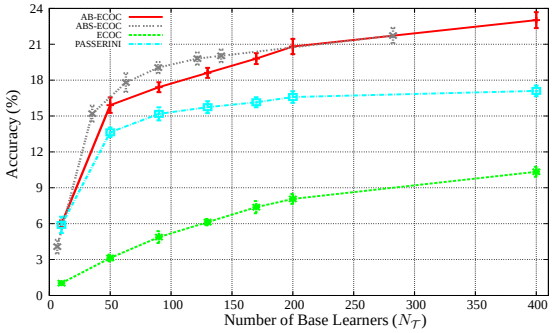
This behavior is closely related to the curse of dimensionality, and most papers in the literature only show the performance going up to 30 classes which is not useful for large-scale problems. Here, we validate our approach for up to 1,000 classes.

4.2 Scenario 2 (200 and 1,000 Classes)

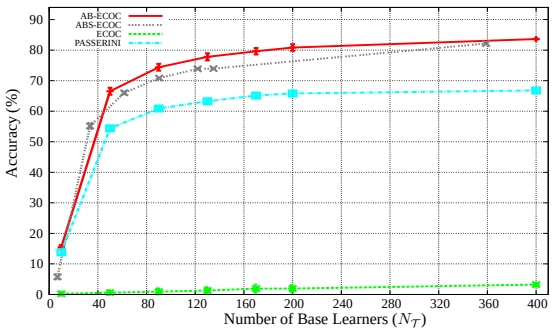
Here, we consider two large-scale Vision applications: Corel ($N_c = 200$) and ALOI ($N_c = 1,000$) categorization. In such applications, OVO is computationally expensive. Sometimes, it is not advised to use OVA at all, given that, even in this case, the number of dichotomies and the number of elements to train are too high. Hence, ECOC approaches with a few base learners are more appropriate. In Figures 2(a–c), we show results using *Affine-Bayes* (AB-ECOC) *vs.* ECOC Hamming decoding and Passerini et al. (Passerini et al., 2004) approaches for LDA and SVM classifiers.

We show experiments up to 400 dichotomies in the presence of 200 and 1,000 classes to 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 classifiers, all approaches fare steadily better, and as this number approaches the limit, they converge to similar results. For ALOI, the limit is $\binom{1,000}{2} = 499,500$, much more than the 400 we show.

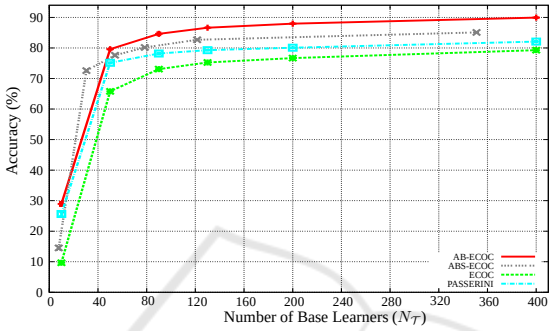
As the image descriptor is not our focus in this pa-



(a) Corel Gallery .:. Base-learner = LDA.



(b) ALOI .:. Base-learner = LDA.



(c) ALOI .:. Base-learner = SVM.

Figure 2: Affine-Bayes (AB) and Affine-Bayes-Shrinking (ABS) *vs.* ECOC *vs.* Passerini for two large-scale data sets.

per, we have used a simple extended color histogram with 128 dimensions (Stehling et al., 2002). Corel data set comprises broad-class images and it is more difficult to classify than the ALOI collection of controlled objects.

Affine-Bayes improves the effectiveness in the two data sets regardless the base learner (LDA or SVM). In both cases and for both data sets, we see that *Affine-Bayes* provides better results than Passerini's and other approaches. For ALOI and SVM base learner, the difference is above 15 standard deviations ($\sim$7–8 percentual points) with respect to Passerini's results.

In addition, *Affine-Bayes* with the shrinking phase provides good results even with fewer dichotomies. For instance, when we provide 200 dichotomies in the training for ALOI data set, *Affine-Bayes* (AB) provides an average accuracy of 80% while *Affine-Bayes-Shrinking* (ABS) provides 76% using only 135 dichotomies. For Corel, when we use AB with 90 dichotomies, the accuracy is 17% while for ABS it is 18%. Finally, in spite of the reduction in the number of dichotomies, *Affine-Bayes* still provides better effectiveness than previous approaches.

For more than 30 classes, the independence restriction play an important role. See Figure 2(b-c). By not assuming independence, the SVM with only 200 dichotomies is more effective than the best K-Nearest neighbors (not shown in the plots). KNN yields $\approx$ 83% accuracy while our approach using SVM and 200 dichotomies yields $\approx$ 88%. We can improve even more if we use 400 dichotomies, and yet, this is much less dichotomies than a solution using one versus all or all combinations of one versus one.

5 CONCLUSIONS AND REMARKS

In this paper, we have addressed two key issues of multi-class classification: the choice of the coding matrix and the decoding strategy. For that, we have presented a new Bayesian treatment for the decoding strategy: *Affine-Bayes*.

We have introduced the concept of affine relations among binary classifiers and presented a principled way to find groups of high correlated base learners. Furthermore, we have presented a strategy to reduce the number of required dichotomies in the multi-class process.

The advantages of our approach are: (1) it works independent of the number of selected dichotomies; (2) it can be associated with each one of the previ-

ous approaches such as OVO, OVA, ECOC, and their combinations; (3) it does not rely on the independence restriction among all dichotomies. (4) its implementation is simply and it uses only basic probability theory; (5) it is fast and does not impact the multi-class procedure.

Future work include the deployment of better policies to choose the coding matrix and the design of alternative ways to store the conditional probability tables other than sparse matrices and hashes.

ACKNOWLEDGEMENTS

We thank FAPESP (Grant 05/58103-3), and CNPq (Grants 309254/2007-8 and 551007/2007-9) for the financial support.

REFERENCES

Allwein, E., Shapire, R., and Singer, Y. (2000). Reducing multi-class to binary: A unifying approach for margin classifiers. *JMLR*, 1(1):113–141.

Anand, R., Mehrotra, K., Mohan, C., and Ranka, S. (1995). Efficient classification for multi-class problems using modular neural networks. *TNN*, 6(1):117–124.

Athisos, V., Stefan, A., Yuan, Q., and Sclaroff, S. (2007). Classmap: Efficient multiclass recognition via embeddings. In *ICCV*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, 1 edition.

Clark, P. and Boswell, R. (1991). Rule induction with CN2: Some improvements. In *EWSL*, pages 151–163.

Crammer, K. and Singer, Y. (2002). On the learnability and design of output codes for multi-class problems. *JMLR*, 47(2–3):201–233.

Dietterich, T. and Bakiri, G. (1996). Solving multi-class problems via ECOC. *JAIR*, 2(1):263–286.

Klautau, A., Jevtic, N., and Orlitsky, A. (2004). On nearest-neighbor ECOC with application to all-pairs multi-class support vector machines. *JMLR*, 4(1):1–15.

Moreira, M. and Mayoraz, E. (1998). Improved pairwise coupling classification with correcting classifiers. In *ECML*.

Narasimhamrthy, A. (2005). Theoretical bounds of majority voting performance for a binary classification problem. *TPAMI*, 27(12):1988–1995.

Passerini, A., Pontil, M., and Frasconi, P. (2004). New results on error correcting output codes of kernel machines. *TNN*, 15(1):45–54.

Pedrajas, N. and Boyer, D. (2006). Improving multi-class pattern recognition by the combination of two strategies. *TPAMI*, 28(6):1001–1006.

Platt, J., Christiani, N., and Taylor, J. (1999). Large margin dags for multi-class classification. In *NIPS*, pages 547–553.

Pujol, O., Radeva, P., and Vitria, J. (2006). Discriminant ECOC: A heuristic method for application dependent design of ECOC. *TPAMI*, 28(6):1007–1012.

Stehling, R., Nascimento, M., and Falcão, A. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109.

Takenouchi, T. and Ishii, S. (2007). Multi-class classification as a decoding problem. In *FOCI*, pages 470–475.

Windeatt, T. and Ghaderi, R. (2003). Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21.

Young, C., Yen, C., Pao, Y., and Nagurka, M. (2006). One-class-at-time removal sequence planning method for multi-class problems. *TNN*, 17(6):1544–1549.

APPENDIX

Table 2: List of useful symbols.

X, x	Data samples, and an element of X.
Y, y	The class' labels of X and an element of Y.
N	Number of elements of X.
N_c	Number of classes.
N_d	The dimensionality X.
C_L, c	The class labels and a class such that $c_i \in C_L$.
Ω	All possible dichotomies for C.
$\mathcal{T}$	A team of dichotomies such that $\mathcal{T} \subset \Omega$.
$N_\mathcal{T}$	The number of dichotomies in $\mathcal{T}$.
M	A coding matrix
$\mathcal{O}_\mathcal{T}$	A realization of T.
$\mathcal{A}$	The affine matrix.
$\mathcal{G}_\mathcal{D}$	The groups of affine dichotomies.
g_i	Group of affine dichotomies such that $g_i \subset \mathcal{G}_\mathcal{D}$.