Voting using Minimally-Sized Decomposition Schemes

Evgueni Smirnov[1], Matthijs Moed[1], Georgi Nalbantov[2], and Ida Sprinkhuizen-Kuyper[3]

[1] Department of Knowledge Engineering, Maastricht University,
P.O.BOX 616, 6200 MD Maastricht, The Netherlands
{smirnov,m.moed}@maastrichtuniversity.nl

[2] Faculty of Health, Medicine and Life Sciences, Maastricht University, P.O.BOX 616, 6200 MD Maastricht, The Netherlands. g.nalbantov@maastrichtuniversity.nl

[3] Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, 6525 HR Nijmegen, The Netherlands. i.kuyper@donders.ru.nl

Abstract. Error-correcting output coding (ECOC) can significantly improve generalization performance on multi-class classification problems by introducing code redundancy. However the code redundancy can cause a computational-complexity problem when the number of classes is large. In this paper we try to avoid this problem. We study a particular class of minimally-sized ECOC decomposition schemes, namely the class of minimally-sized balanced decomposition schemes (MBDSs) [11]. We show that MBDSs do not face a computational-complexity problem for large number of classes but cannot correct the classification errors of the binary classifiers they define. Therefore we propose voting with MBDS classifiers (VMBDSs). We show that the generalization performance of the VMBDSs ensembles improves with the number of MBDS classifiers. However this number can become large and thus the VMBDSs ensembles can have a computational-complexity problem as well. Fortunately our experiments show that VMBDSs are comparable with ECOC ensembles and can outperform one-against-all ensembles using only a small number of MBDS classifiers.

1 Introduction

A decomposition scheme for a multi-class classification problem is a mapping from a class set Y to a set of binary partitions of Y [4, 10]. It represents that problem by a set of binary classification problems. The decomposition scheme is applied on two stages [4, 10]: encoding and decoding. During the encoding stage we generate binary classification problems following the decomposition scheme and train a binary classifier for each problem. During the decoding stage we apply the binary classifiers for a test instance and combine their class predictions to estimate the instance class.

The main family of decomposition schemes is that of error-correcting output coding (ECOC) [4]. Due to code redundancy, ECOC can significantly improve generalization performance on multi-class classification problems. However the code redundancy can cause a computational-complexity problem for ECOC: the number of binary classification problems can grow exponentially with the number of classes. Several approaches to this problem were proposed [1, 4, 14]. In essence these approaches try to maximize the diversity of the binary partitions in ECOC schemes for a fixed scheme size.

In this paper we address the computational-complexity problem of ECOC schemes as well. For that purpose we study ECOC decomposition schemes of minimal size. In this respect our research differs from previous studies which so far have only focused on ECOC schemes of a fixed size [1, 4, 14]. In the paper we show that minimally-sized ECOC decomposition schemes can be viewed as minimally-sized balanced decomposition schemes (MBDSs). We note that MBDSs were suggested in [11] but was never studied in detail. This paper provides a deep analysis of MBDSs. First, we prove that the size of MBDSs equals $\lceil \log_2(|Y|) \rceil$. This property implies that the MBDSs ensembles do not have a computational-complexity problem and thus can be used for classification problems with a large number of classes. Second, we quantify the space of all possible MBDSs. Third, we analyze the error-correction properties of MBDSs. We show that: (a) the minimal Hamming distance between MBDS class code words equals 1, and (b) the Hamming distance between MBDS class-partition code words equals $\frac{|Y|}{2}$. Thus MBDSs cannot correct the classification errors of the binary classifiers in MBDS ensembles.

To enforce error correction we propose voting with MBDS ensembles (VMBDSs). We show that the VMBDSs ensembles improve generalization performance with the number of MBDS classifiers. However this number can be large and the VMBDSs ensembles can have a computational-complexity problem. Fortunately our experiments demonstrate that VMBDSs are comparable with ECOC ensembles and can outperform one-against-all ensembles using a small number of MBDS ensembles.

The paper is organized as follows. Section 2 formalizes the classification problem. Decomposing multi-class classification problems is discussed in Section 3. Section 4 introduces minimally-sized balanced decomposition schemes and voting based on these schemes. Experiments are given in Section 5. Section 6 concludes the paper.

2 Classification Problem

Let X be an instance space and Y be a class set of size K greater than 1. An instance is a tuple (x, y) where $x \in X, y \in Y$. Training data D is a set of instances. Given the training data D, the classification problem CP is to find the class of an instance $x \in X$. When $K = 2$, the classification problem is a *binary classification problem BCP*. When $K > 2$, the classification problem is a *multi-class classification problem MCP*. Many classifiers are binary for various reasons. However, many real-world classification problems are *multi-class* problems. There are two approaches for solving a multi-class problem using a binary classifier: direct and indirect. The direct approach is to generalize the binary classifier to a multi-class classifier (e.g., support vector machines). The indirect approach is to employ *decomposition schemes* (e.g., ECOC [4, 10]) considered below.

3 Decomposing Multi-Class Classification Problems

This section considers the main elements needed for decomposing a multi-class classification problem into a set of binary problems: the decomposition scheme, the encoding stage, and the decoding stage. Subsection 3.1 formalizes the concept of *decomposition scheme* and presents two of the most well-known decomposition schemes. Subsection 3.2 provides a detailed explanation of *the encoding and decoding stages*.

3.1 Decomposition Schemes

Consider a multi-class classification problem MCP determined on a class set Y of size $K > 2$. To show how to decompose MCP into L binary classification problems BCP_l we define the notion of a *binary class partition* in Definition 1.

Definition 1. (Binary Class Partition) *Given a class set Y, the set $P(Y)$ is a binary class partition of Y iff $P(Y)$ equals $\{Y^+, Y^-\}$ s.t. $Y^- \cup Y^+ = Y$ and $Y^- \cap Y^+ = \emptyset$.*

Definition 1 allows us to introduce the notion of a *decomposition scheme*. A decomposition scheme describes how to decompose a multi-class classification problem MCP into L binary classification problems BCP_l, as given in Definition 2.

Definition 2. (Decomposition Scheme) *Given multi-class classification problem MCP and positive integer L, the* decomposition scheme *of MCP is a set $SP(Y)$ of L binary class partitions $P_l(Y)$ s.t. for any two classes $y_1, y_2 \in Y$ there exists a binary class partition $P_m(Y) \in SP(Y)$ s.t. $\neg(y_1, y_2 \in Y_m^-) \wedge \neg(y_1, y_2 \in Y_m^+)$ where $Y_m^-, Y_m^+ \in P_m(Y)$.*

Any decomposition scheme $SP(Y)$ has L binary class partitions $P_l(Y)$. The partitions $P_l(Y) \in SP(Y)$ are chosen s.t. any class $y \in Y$ is determined uniquely.

A natural representation for a decomposition scheme $SP(Y)$ is a *decomposition matrix M*. M is a binary matrix $\{-1, +1\}^{K \times L}$ encoded according to the following rule:

$$M_{k,l} = \begin{cases} -1 & \text{if class } y_k \in Y \text{ belongs to } Y_l^- \text{ of } P_l(Y); \\ +1 & \text{if class } y_k \in Y \text{ belongs to } Y_l^+ \text{ of } P_l(Y). \end{cases}$$

The rows of M form *class code words w_{y_k}* corresponding to the K classes in the set Y. The columns of M form *class-partition code words $w_{P_l(Y)}$* corresponding to the L binary class partitions $P_l(Y)$. Decomposition matrices have column and row properties that follow from Definition 2. These properties are formulated in Corollary 3.

Corollary 3. *Consider decomposition matrix M. Then any two columns $M_{*,l}, M_{*,m} \in M$ are different if $l \neq m$ and any two rows $M_{k,*}, M_{o,*} \in M$ are different if $k \neq o$.*

Any two decomposition matrices are equivalent if they represent the same decomposition scheme.

Theorem 4. *For any decomposition scheme $SP(Y)$ there exist $L! \times 2^L$ equivalent decomposition matrices M.*

Two most well-known decomposition schemes are “one-against-all”(OA) [12] and “exhaustive error-correcting output coding” (eECOC) [4]. OA is a decomposition scheme consisting of all possible binary class partitions containing a class set with size 1. Hence any OA decomposition matrix has dimensions $K \times K$. eECOC is a decomposition scheme [4] consisting of all possible binary class partitions. This implies that the eECOC decomposition scheme is a superset of the OA decomposition scheme. The number of binary class partitions in the eECOC decomposition scheme equals $2^{K-1} - 1$.

3.2 Encoding and Decoding

To solve a multi-class classification problem using a decomposition scheme $SP(Y)$ we need to pass two stages: *encoding* and *decoding*. Below we describe these stages.

During the encoding stage we first generate binary classification problems BCP_l according to a given decomposition scheme $SP(Y)$. Each BCP_l is uniquely determined by a particular binary class partition $P_l(Y) \in SP(Y)$. BCP_l is defined on the instance space X and a class set given by the binary class partition $P_l(Y)$. The training data D_l for BCP_l consists of instances $(x, Y_l^{\pm}) \in X \times P_l(Y)$ and for any instance $(x, Y_l^{\pm}) \in D_l$ there exists an instance (x, y) from the training data D of the multi-class classification problem MCP s.t. $y \in Y_l^{\pm}$. Thus, the decomposition scheme $SP(Y)$ reduces the multi-class classification problem MCP to L binary classification problems BCP_l.

Once the binary classification problems BCP_l have been determined, we train a binary classifier $h_{P(Y)} : X \rightarrow P(Y)$ for each BCP_l. The binary classifiers $h_{P(Y)}$ together form an ensemble classifier $h_{SP(Y)} : X \rightarrow Y$ equal to $\{h_{P(Y)}\}_{P(Y) \in SP(Y)}$.

During the decoding stage, given a test instance $x \in X$ and an ensemble classifier $h_{SP(Y)}$, we need to decode the predictions provided by the binary classifiers $h_{P(Y)} \in h_{SP(Y)}$ to form a class estimate $y \in Y$ for x. The OA and eECOC decomposition schemes both use the same decoding technique. This decoding technique first takes the class score $S(x, y|h_{P(Y)})$ provided by each binary classifier $h_{P(Y)} \in h_{SP(Y)}$ (see Definition 5 below) and then computes the final score $S(x, y|h_{SP(Y)})$ of the ensemble classifier $h_{SP(Y)}$ as the sum of scores $S(x, y|h_{P(Y)})$ over all the classifiers $h_{P(Y)} \in h_{SP(Y)}$ (see Definition 6 below).

Definition 5. *Given a binary class partition $P(Y) \in SP(Y)$, a binary classifier $h_{P(Y)} : X \rightarrow P(Y)$, a test instance $x \in X$ and a class $y \in Y$, the class score $S(x, y|h_{P(Y)})$ for x and y provided by $h_{P(Y)}$ is defined as follows:*

$$S(x, y|h_{P(Y)}) = \begin{cases} 1 & \text{if class } y \in h_{P(Y)}(x); \\ 0 & \text{if class } y \notin h_{P(Y)}(x). \end{cases}$$

Definition 6. *Given a decomposition scheme $SP(Y)$, an ensemble classifier $h_{SP(Y)}$, a test instance $x \in X$ and a class $y \in Y$, the total class score $S(x, y|h_{SP(Y)})$ for x and y provided by $h_{SP(Y)}$ equals $\sum_{P(Y) \in SP(Y)} S(x, y|h_{P(Y)})$.*

Traditionally, decoding is explained using decomposition matrices [4]. In this case for any test instance $x \in X$ the predictions $h_{P(Y)}(x) \in h_{SP(Y)}$ provided by the binary classifiers are first mapped to either -1 or +1 according to the column of a decomposition matrix M corresponding to $P(Y)$. Then the resulting numbers are combined into a class code word $\hat{w}$ according to the order of the columns in M. This class code word $\hat{w}$ is compared against each class code word in M and the instance x is assigned to a class whose code word is closest according to the Hamming distance.

A decomposition matrix M has to satisfy two properties [13]:

- **Row separation**: any class code word $M_{k,*}$ in M should be well-separated from all other class code words $M_{m,*}$ in terms of Hamming distance.

- **Column separation**: any class-partition code word $M_{*,l}$ in M should be well-separated from many other class-partition code words $M_{*,n}$ and their complements $-M_{*,n}$ in terms of Hamming distance.

The first property implies that the ensemble classifier $h_{SP(Y)}$ is capable of correcting the errors of $\lfloor \frac{H_{min}-1}{2} \rfloor$ binary classifiers $h_{P(Y)} \in h_{SP(Y)}$ where H_{min} is the minimum Hamming distance between class code words in the decomposition matrix M. The second property aims at minimally correlating the errors of the binary classifiers $h_{P(Y)} \in h_{SP(Y)}$, thus minimizing the number of the classifiers $h_{P(Y)}$ which predictions have to be corrected. Increasing the Hamming distance of class code words can be achieved by introducing additional code redundancy; i.e., by increasing the number L of columns in M. In the extreme case of the eECOC decomposition scheme this leads to L equal to $2^{K-1}-1$. Thus, code redundancy can cause a computational-complexity problem for ECOC schemes: the number of binary classification problems can grow exponentially with the number of classes. Several approaches to this problem of ECOC were proposed [1, 4, 14]. In essence they try to design ECOC schemes that maximize the minimum Hamming distance between class code words and class-partition code words for a fixed scheme size L.

4 (Minimally-Sized) Balanced Decomposition Schemes

This Section addresses the computational-complexity problem of ECOC. In contrast with the previous work it focuses on ECOC schemes of minimal size. It shows that these schemes belong to the class of balanced decomposition schemes. Therefore, Subsection 4.1 first introduces balanced decomposition schemes. Then Subsection 4.2 studies minimally-sized ECOC schemes considered as minimally-sized balanced decomposition schemes (MBDSs). Finally in Subsection 4.3 we propose voting based on MBDSs.

4.1 Balanced Decomposition Schemes

Balanced decomposition schemes are a subclass of decomposition schemes based on balanced binary class partitions. The balanced binary class partitions are defined below.

Definition 7. (Balanced Binary Class Partitions) *If the number K of classes in a class set Y is even, then a binary class partition $P(Y) = \{Y^-, Y^+\}$ is balanced iff $|Y^-| = |Y^+|$.*

Corollary 8. *The number of all balanced binary class partitions $P(Y)$ equals $\frac{K!}{2(\frac{K}{2}!)^2}$.*

Given the concept of balanced binary class partitions we define the concept of *balanced decomposition schemes* in Definition 9 below.

Definition 9. (Balanced Decomposition Schemes) *A decomposition scheme $SP(Y)$ is said to be* balanced *iff each binary class partition $P(Y) \in SP(Y)$ is balanced.*

Theorem 10. *The class of balanced decomposition schemes is non-empty.*

Balanced decomposition schemes do not introduce additional imbalance in the training data of the binary classifiers $h_{P(Y)}$ in the ensemble $h_{SP(Y)}$. In this respect balanced decomposition schemes differ from other decomposition schemes (e.g., OA, eECOC).

4.2 Minimally-Sized Balanced Decomposition Schemes

Minimally-sized balanced decomposition schemes (MBDSs) are balanced decomposition schemes of minimal size. This type of decomposition schemes was given in [11] but was never studied in detail. This subsection provides the definition and properties of MBDSs.

Definition 11. (Minimally-Sized Balanced Decomposition Schemes) *Given the set $SP^M(Y)$ of all balanced binary class partitions, a balanced decomposition scheme $SP(Y) \subseteq SP^M(Y)$ is said to be minimally-sized iff there does not exist another balanced decomposition scheme $SP'(Y) \subseteq SP^M(Y)$ s.t. $|SP'(Y)| < |SP(Y)|$.*

Notation 1. A minimally-sized balanced decomposition scheme is denoted by $SP^m(Y)$.

Theorem 12 determines the size of MBDSs as a function of the number K of classes.

Theorem 12. *A balanced decomposition scheme $SP(Y)$ is minimally-sized iff the size of $SP(Y)$ equals $\lceil \log_2(K) \rceil$.*

Corollary 13. *Any decomposition matrix M of a minimally-sized balanced decomposition scheme $SP(Y)^m$ forms a minimally-sized binary code for K classes.*

For a multi-class classification problem we can define different MBDSs.

Theorem 14. *If K is a power of 2, the number of all possible minimally-sized balanced decomposition schemes $SP^m(Y)$ equals $\frac{(K-1)!}{\log_2(K)!}$.*

The decoding stage for the MBDSs is realized according to Definition 6. In this context we determine the (minimal) Hamming distance of class code words and class-partition code words in decomposition matrices of MBDSs in Corollary 15.

Corollary 15. *If the number K of classes is a power of 2, then for any decomposition matrix M of a minimally-sized balanced decomposition scheme $SP^m(Y)$:*

(1) the minimal Hamming distance between different rows $M_{k,*}, M_{o,*} \in M$ equals 1,
(2) the Hamming distance between different columns $M_{*,l}, M_{*,m} \in M$ equals $\frac{K}{2}$.

The results from Corollary 15 directly imply that:

- **The row separation property** does not hold for MBDSs, since the minimum Hamming distance between class code words in any decomposition matrix of a MBDS equals one. Thus, the ensemble classifiers $h_{SP(Y)}$ based on these decomposition schemes are not capable of correcting errors of the binary classifiers $h_{P(Y)} \in h_{SP(Y)}$.
- **The column separation property** does hold for the MBDSs, since the Hamming distance between the class-partition code words in any decomposition matrix of a MBDS equals $\frac{K}{2}$ (the maximal possible distance). Thus, we expect that the errors of the binary classifiers $h_{P(Y)} \in h_{SP(Y)}$ are minimally correlated.

From the above we conclude that MBDSs have an error-correction problem; i.e., MBDSs do not have error-correction capabilities. Nevertheless, when the number of classes is very large, MBDSs can be a viable alternative to the eECOC decomposition scheme. This is due to the fact that they do not have a computational-complexity problem. We note that by Theorem 12 the number of the binary classifiers $h_{P(Y)}$ in any MBDS ensemble $h_{SP(Y)}$ equals $\lceil \log_2(K) \rceil$.

4.3 Voting using Minimally-Sized Balanced Decomposition Schemes

To avoid the error-correction problem of MBDSs we propose to vote with ensemble classifiers $h_{SP(Y)}$ based on MBDSs. This approach is called Voting using Minimally-Sized Balanced Decomposition Schemes (VMBDSs) and it is considered below.

Let $SSP(Y)$ be a set of N randomly-chosen minimally-sized balanced decomposition schemes $SP^m(Y)$. Each minimally-sized balanced decomposition scheme $SP^m(Y)$ defines a classifier $h_{SP^m(Y)}$. The classifiers $h_{SP^m(Y)}$ form an ensemble classifier $h_{SSP(Y)}$: $X \to Y$ equal to $\{h_{SP^m(Y)}\}_{SP^m(Y) \in SSP(Y)}$. Decoding the predictions of the classifiers $h_{SP^m(Y)}$ into the prediction of $h_{SSP(Y)}$ is realized for any test instance $x \in X$ and class $y \in Y$ by computing an integer score $S(x,y|h_{SSP(Y)})$. This computation is a two-stage process: first we take the class score $S(x,y|h_{SP^m(Y)})$ provided by each classifier $h_{SP^m(Y)}$ (see Definition 6) and then compute the score $S(x,y|h_{SSP(Y)})$ as the sum of scores $S(x,y|h_{SP^m(Y)})$ over all the classifiers $h_{SP^m(Y)}$ (see Definition 16).

Definition 16. *Given a set $SSP(Y)$ of minimally-sized balanced decomposition schemes $SP^m(Y)$, a test instance $x \in X$, and a class $y \in Y$, the class score $S(x,y|h_{SSP(Y)})$ for x and y provided by the ensemble classifier $h_{SSP(Y)}$ equals $\sum_{SP^m(Y) \in SSP(Y)} S(x,y|h_{SP^m(Y)})$.*

The score $S(x,y|h_{SSP(Y)})$ can be further refined by combining definitions 6 and 16:

$$S(x,y|h_{SSP(Y)}) = \sum_{SP^m(Y) \in SSP(Y)} \sum_{P(Y) \in SP^m(Y)} S(x,y|h_{P(Y)}). \tag{1}$$

Thus, the class with the highest score $S(x,y|h_{SSP(Y)})$ will be the class $y \in Y$ that receives most of the votes $S(x,y|h_{P(Y)})$ of the binary classifiers $h_{P(Y)}$.

The class with the highest score according to Equation 1 can be determined by Hamming decoding as well. For that purpose we consider a decomposition matrix $M_{SSP(Y)}$ with dimensions $K \times N \log_2(K)$. The matrix consists of the class-partition code words of the decomposition matrices of the minimally-sized balanced decomposition schemes $SP^m(Y) \in SSP(Y)$ given some order over the class set Y. Classifying any instance $x \in X$ using the decomposition matrix $M_{SSP(Y)}$ is realized using the standard decoding procedure described in Subsection 3.2. It is easy to show that the final class for the instance x is exactly that which maximizes the score given in Equation 1.

Since VMBDSs can be explained using the decomposition matrix $M_{SSP(Y)}$, we analyze the properties of class code words and class-partition code words in $M_{SSP(Y)}$.

- **Class code words:** the Hamming distance between class code words in $M_{SSP(Y)}$ is computed for non-repeated columns only. Hence, if we have two class code words $M_{i,*}, M_{j,*} \in M_{SSP(Y)}$ that differ in positions o and p, their Hamming distance equals 2 if class-partition code words $M_{*,o}, M_{*,p} \in M_{SSP(Y)}$ are different; otherwise, it is equal to 1. In the case when all minimally-sized balanced decompositions $SP^m(Y) \in SSP(Y)$ are disjointed (i.e., no column repetition in $M_{SSP(Y)}$), the minimal Hamming distance between class code words in $M_{SSP(Y)}$ equals N.
- **Class-partition code words:** the Hamming distance between any two class-partition code words in the decomposition matrix $M_{SSP(Y)}$ decreases. If minimally-sized balanced decompositions $SP^m(Y) \in SSP(Y)$ are not disjointed, then the minimal Hamming distance between class-partition code words that belong to different

MBDSs is in the range $[0, \frac{K}{2}]$; otherwise, it is in the range $[2, \frac{K}{2}]$. In both cases the errors of the binary classifiers $h_{P(Y)}$ that belong to different classifiers $h_{SP(Y)} \in h_{SSP(Y)}$ can be more correlated compared with a MBDS ensemble.

From the above it follows that the row separation and column separation properties hold for VMBDSs iff minimally-sized balanced decompositions $SP^m(Y) \in SSP(Y)$ are disjointed. Thus, if we have a VMBDSs ensemble with N MBDS classifiers, we can correct the errors of $\lfloor \frac{N-1}{2} \rfloor$ binary classifiers $h_{P(Y)} \in \bigcup_{SP^m(Y) \in SSP(Y)} h_{SP^m(Y)}$.

We note that in the extreme case VMBDSs ensembles can contain $\frac{(K-1)!}{\log_2(K)!}$ MBDS classifiers (see Theorem 14). Thus, the VMBDSs ensembles can have a computational-complexity problem. Fortunately, our experiments show that the accuracy of MBDS ensembles is already close to that of eECOC ensembles for small N.

5 Experiments

To assess the generalization performance of MBDS and VMBDSs ensembles we performed two sets of experiments. The first set of experiments, provided in Section 5.1, compares the classification accuracy of MBDS and VMBDSs ensembles against that of eECOC and OA ensembles on 15 UCI datasets [2]. The second set of experiments, discussed in Section 5.2, compares the classification accuracy of VMBDSs ensembles against OA ensembles on data sets with a large number of classes.

5.1 UCI Data Experiments

The purpose of the experiments in this section is to compare the accuracy of MBDS, VMBDSs, eECOC, and OA ensembles on 15 UCI datasets [2]. Three types of classifiers were employed as binary base classifiers: the Ripper rule classifier [3], logistic regression [7], and support vector machines [5]. The number of MBDS classifiers in the VMBDSs ensembles varied from 1 to 15. The evaluation method was 10-fold cross validation averaged over 10 runs. The results are given in Table 3, Table 4, and Table 5 in the Appendix. The classification accuracy of the classifiers was compared using the corrected paired t-test at the 5% significance level. Two types of t-test comparisons were realized: eECOC ensembles against all other ensembles and OA ensembles against all other ensembles. An analysis of the results in Tables 3-5 show that:

- The MBDS ensembles are the worst ensembles. The experiments confirm that the MBDS ensembles do not have error-correction capabilities.
- The VMBDSs ensembles perform much better than MBDS ensembles. Their classification accuracy improves with the number of the MBDS classifiers. The results confirm that the VMBDSs ensembles do have error-correction capabilities.
- The VMBDSs ensembles are comparable with the eECOC ensembles in terms of classification accuracy if the number of MBDS ensembles is more than one. In this case VMBDSs ensembles are preferable, since they need less binary classifiers.
- The VMBDSs ensembles can outperform the OA ensembles in terms of classification accuracy if the number of the MBDS ensembles is greater than two.

Table 1: Number of instances and classes for the Abalone, Patents, and Faces94 datasets.

Name	**#instances**	**#classes**
Abalone	4177	28
Patents	2373	70
Faces94	3059	153

Table 2: Classification Accuracy of SVM-based OA and VMBDSs ensembles on the Abalone, Patents, and Faces94 datasets. The numbers in the VMBDSs columns indicate the number of MBDS classifiers in the VMBDSs ensemble. Bold numbers indicate statistically better results with respect to the OA ensembles.

		VMBDSs		
Data set	**OA**	**5**	**10**	**25**
Abalone	0.023 ± 0.002	**8.65 ± 5.31**	**13.53 ± 4.58**	**18.05 ± 4.29**
Patents	17.23 ± 1.05	19.52 ± 1.81	**21.13 ± 1.71**	**21.54 ± 1.30**
Faces94	73.85 ± 4.29	74.98 ± 2.06	**87.26 ± 1.21**	**93.68 ± 0.80**

5.2 Experiments on Data Sets with Large Number of Classes

The purpose of this section's experiments is to compare the classification accuracy of the VMBDSs and OA ensembles on three datasets with a large number of classes. The datasets chosen are Abalone [2], Patents [9], and Faces94 [8]. Several properties of these datasets are summarized in Table 1.

The eECOC ensembles were excluded from the experiments, since they require an exponential number of binary classifiers (at least $2^{27} - 1$). Support vector machines [5] were used as a base classifier. The number of MBDS classifiers in the VMBDSs ensembles was varied from 5 - 25. The evaluation method was 5-fold cross validation averaged over 5 runs. The results are presented in Table 2. The classification accuracy of the classifiers is compared using the corrected paired t-test at the 5% significance level. The test compares the OA ensembles against all VMBDSs ensembles.

The experimental results from Table 2 show that the VMBDSs ensembles can outperform statistically the OA ensembles on these three datasets. In this respect it is important to know whether the VMBDSs ensembles outperform the OA ensembles when both types of ensembles contain the same number of binary classifiers. This dataset has 28 classes. Thus, the number of binary classifiers in the OA ensemble is 28. This implies that we need a configuration for the VMBDSs ensembles with number of binary classifiers close to 28. In this context we note that the number of binary classifiers in each MBDS ensemble is $\lceil \log_2(28) \rceil = 5$. Thus, in order to have close to 28 number of binary classifiers we need $\lfloor \frac{28}{5} \rfloor = 5$ MBDS classifiers. According to Table 2 for this configuration the VMBDSs ensemble outperforms statistically the OA ensemble. Analogously we can do the same computation for the Patents and Faces94 datasets: for the Patents dataset we need 10 MBDS classifiers and for the Faces94 dataset we need 19 MBDS classifiers in the VMBDSs ensemble. According to Table 2 for these configurations the VMBDSs ensembles outperform statistically the OA ensemble.

6 Conclusion

In this paper we addressed the computational-complexity problem of the ECOC decomposition schemes. We provided a deep analysis of the MBDSs schemes . We proved that the size of MBDSs equals $\lceil \log_2(|Y|) \rceil$. This property implies that MBDSs do not have a computational-complexity problem for large number of classes. We quantified the space of all possible MBDSs. We analyzed the error-correction properties of MBDSs and showed that the minimal Hamming distance between MBDS class code words equals 1. Thus, we concluded that MBDSs cannot correct the classification errors of the binary classifiers in MBDS ensembles. To enforce error correction we proposed voting with MBDS ensembles (VMBDSs). We showed that VMBDSs improve generalization performance with the number of MBDS classifiers. However this number can be large and the VMBDSs ensembles can have a computational-complexity problem. Fortunately our experiments demonstrated that the VMBDSs ensembles are comparable with the ECOC ensembles and can outperform the one-against-all ensembles for a small number of the MBDS classifiers.

The practical value of the VMBDSs ensembles stems from the fact that their generalization performance is comparable with that of ECOC schemes and that VMBDSs ensembles require a smaller number of binary classifiers. In practice designing VMBDSs ensembles can be realized as follows. First we estimate the time needed to train one binary classifier. Then we use this time to estimate the time to train one MBDS ensemble. Finally we decide how many MBDS ensembles need to be included in the resulting VMBDSs ensemble depending on the time restriction imposed.

References

1. Allwein, E., Schapire, R.: Singer, Y., Kaelbling, Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
2. Asuncion, A., Newman, D.J.: UCI machine learning repository (2011) http://www.ics.uci.edu/∼mlearn/MLRepository.html
3. Cohen, W.: Fast effective rule induction. In: Prieditis, A., Russell, R. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 115-123. Morgan Kaufmann (1995)
4. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
5. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation 13, 637–649 (2001)
6. Kong, E.B., Dietterich, T.G.: Error-correcting output coding corrects bias and variance. In: Prieditis, A., Russell, S.J. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 313–321. Morgan Kaufmann (1995)
7. le Cessie, S., van Houwelingen, J.C.: Ridge estimators in logistic regression. Applied Statistics 41, 191–201 (1992)
8. Libor, S.: Face recognition database (2011) http://cswww.essex.ac.uk/mv/allfaces/index.html
9. Lissoni, F., Llerena P., Sanditov, B.: Inventors small worlds: academic and CNRS researchers in networks of inventors in France. In: Proceedings of the DIME Final Conference, 6-8 April 2011, Maastricht, The Netherlands (2010)

10. Lorena, A.C., De Carvalho, A., Joo, M. P.: A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review 30, 19–37 (2008)
11. Mayoraz, E., Moreira, M.: On the decomposition of polychotomies into dichotomies. In: Fisher, D.H. (ed.) Proceedings of the Fourteenth International Conference on Machine Learning, pp. 219–226. Morgan Kaufmann (1996)
12. Nilsson, N.: Learning Machines: Foundations of Trainable Pattern-Classifying Systems, McGraw-Hill New York, (1965)
13. Peterson, W., Weldon J.: Error-Correcting Codes, MIT Press Cambridge, (1972)
14. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141 (2004)
15. Witten, I., Frank,E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, (2011)

APPENDIX: Classification Accuracy of eECOC, OA, and VMBDSs Ensembles on 15 UCI Datasets

This appendix contains the experimental results in terms of the classification accuracy of the eECOC, OA, and VMBDSs ensembles based on three base classifiers: Ripper (Table 3), logistic regression (Table 4), and support vector machines (Table 5). The table header numbers in the VMBDSs columns show the number of MBDS classifiers in the VMBDSs ensembles. The numbers after the data-set names indicate the number of classes. The lower-left (upper-left) dots show statistically worse (better) results with respect to the OA ensembles. The lower-right (upper-right) dots indicate statistically worse (better) results with respect to the eECOC ensembles. The statistical test is the corrected paired t-test at the 5% significance level.

Table 3: Classification accuracy of the OA, eECOC, and VMBDSs ensembles using the Ripper classifier as a base classifier.

			VMBDSs						
Data set	**OA**	**eECOC**	**1**	**2**	**3**	**4**	**5**	**10**	**15**
`car(4)`	88.5	89.5	$_\bullet$83.5$_\bullet$	87.0$_\bullet$	87.2$_\bullet$	n/a	n/a	n/a	n/a
`hypothy(4)`	99.2	99.3	99.1	99.1	99.2	n/a	n/a	n/a	n/a
`lymphog(4)`	77.8	78.7	73.4	75.3	75.2	n/a	n/a	n/a	n/a
`molecul(4)`	28.9	26.9	27.7	26.6	26.7	n/a	n/a	n/a	n/a
`clevela(5)`	79.8	80.4	77.0	78.1	79.2	79.8	80.2	80.2	80.5
`hungari(5)`	79.8	79.9	77.9	78.6	79.2	79.7	79.8	79.8	80.2
`page-bl(5)`	97.0$_\bullet$	$^\bullet$97.4	$_\bullet$96.5$_\bullet$	96.7$_\bullet$	97.0$_\bullet$	97.1	97.2	97.2	97.3
`anneal(6)`	98.7	98.5	97.9	98.1	98.2	98.4	98.3	98.4	98.5
`bridge1(6)`	64.0	66.8	58.4	60.0	62.2	63.4	63.3	64.1	65.0
`bridge2(6)`	64.1	65.6	58.0	59.2	60.3	61.7	62.3	63.3	64.6
`autos(7)`	71.9$_\bullet$	$^\bullet$79.0	67.5$_\bullet$	72.6	75.0	75.8	76.3	77.1	77.4
`glass(7)`	67.2$_\bullet$	$^\bullet$74.9	64.8$_\bullet$	68.5$_\bullet$	70.7	70.5	71.3	73.0	73.7
`zoo(7)`	91.5	93.1	89.0	91.0	91.5	92.4	91.9	92.7	93.0
`ecoli(8)`	80.7$_\bullet$	$^\bullet$85.4	$_\bullet$76.3$_\bullet$	79.8$_\bullet$	82.8	84.1	83.7	83.7	84.1
`flags(8)`	58.3	62.3	51.1$_\bullet$	56.1	58.4	59.3	60.0	61.2	61.1

Table 4: Classification accuracy of the OA, eECOC, and VMBDSs ensembles using logistic regression as a base classifier.

			VMBDSs						
Data set	**OA**	**eECOC**	**1**	**2**	**3**	**4**	**5**	**10**	**15**
car(4)	90.1	89.8	$_\bullet$86.2$_\bullet$	88.9	90.0	n/a	n/a	n/a	n/a
hypothy(4)	95.1$_\bullet$	$^\bullet$95.3	95.3	95.5	95.3	n/a	n/a	n/a	n/a
lymphog(4)	78.4	77.7	77.4	77.9	77.4	n/a	n/a	n/a	n/a
molecul(4)	30.0	29.5	27.9	29.1	28.5	n/a	n/a	n/a	n/a
clevela(5)	83.7	83.7	83.7	83.7	83.7	83.7	83.7	83.7	83.7
hungari(5)	84.2	84.2	84.2	84.2	84.2	84.2	84.2	84.2	84.2
page-bl(5)	95.7$^\bullet$	$_\bullet$95.3	$_\bullet$94.6$_\bullet$	$_\bullet$94.9$_\bullet$	$_\bullet$95.1	$_\bullet$95.2	$_\bullet$95.2	$_\bullet$95.3	$_\bullet$95.3
anneal(6)	99.5	99.6	99.2	99.4	99.5	99.5	99.6	99.7	99.7
bridge1(6)	59.6	63.6	46.6$_\bullet$	50.1$_\bullet$	53.6$_\bullet$	55.9	54.3$_\bullet$	57.1	58.8
bridge2(6)	55.0	57.8	46.7$_\bullet$	50.1$_\bullet$	51.4	52.5	52.3	54.6	54.0
autos(7)	66.9	71.6	$_\bullet$57.9$_\bullet$	62.5$_\bullet$	64.6$_\bullet$	66.3	66.8	68.8	69.8
glass(7)	64.3	63.9	$_\bullet$57.2$_\bullet$	59.7	60.3	60.5	60.0$_\bullet$	62.0	62.4
zoo(7)	89.5	91.4	82.2$_\bullet$	88.0	89.6	89.6	90.5	91.7	91.6
ecoli(8)	86.5	86.0	$_\bullet$76.9$_\bullet$	$_\bullet$82.5$_\bullet$	85.1	86.0	85.8	85.6	85.8
flags(8)	47.1	51.3	$_\bullet$37.0$_\bullet$	42.1$_\bullet$	43.8$_\bullet$	45.8	46.6	47.9	49.0

Table 5: Classification accuracy of the OA, eECOC, and VMBDSs ensembles using SVM as a base classifier.

			VMBDSs						
Data set	**OA**	**eECOC**	**1**	**2**	**3**	**4**	**5**	**10**	**15**
car(4)	81.0$_\bullet$	$^\bullet$86.0	$^\bullet$87.6	$^\bullet$87.1	$^\bullet$86.0	n/a	n/a	n/a	n/a
hypothy(4)	93.3	93.3	93.3	93.4	93.3	n/a	n/a	n/a	n/a
lymphog(4)	85.2	85.6	85.0	84.2	83.0	n/a	n/a	n/a	n/a
molecul(4)	29.0	29.0	28.5	27.9	28.9	n/a	n/a	n/a	n/a
clevela(5)	83.8	83.8	83.8	83.8	83.8	83.8	83.8	83.9	83.9
hungari(5)	82.7	82.7	82.7	82.7	82.7	82.7	82.7	82.7	82.7
page-bl(5)	92.0	92.0	92.1	91.9	92.1	92.2	92.3	92.4	$^\bullet$92.6
anneal(6)	96.6$_\bullet$	$^\bullet$97.3	96.7	96.9	97.2	97.3	97.4	97.5	97.6
bridge1(6)	58.4$_\bullet$	$^\bullet$65.9	61.2	61.5	62.9	63.4	64.8	$^\bullet$66.3	$^\bullet$65.4
bridge2(6)	61.5$_\bullet$	$^\bullet$67.3	64.5	64.6	65.7	66.5	$^\bullet$67.3	66.7	66.7
autos(7)	56.0$_\bullet$	$^\bullet$64.7	58.3	60.2	$^\bullet$62.6	$^\bullet$62.9	$^\bullet$63.4	$^\bullet$63.9	$^\bullet$64.6
glass(7)	44.4$_\bullet$	$^\bullet$51.3	48.7	40.6$_\bullet$	$^\bullet$50.1	$^\bullet$50.0	46.6	51.5	49.9
zoo(7)	93.6	94.7	94.5	94.4	95.0	95.0	95.0	94.7	94.5
ecoli(8)	68.6$_\bullet$	$^\bullet$81.1	$^\bullet$75.4$_\bullet$	$^\bullet$76.9$_\bullet$	$^\bullet$77.7	$^\bullet$79.5	$^\bullet$79.8	$^\bullet$80.1	$^\bullet$80.3
flags(8)	55.1$_\bullet$	$^\bullet$63.3	53.5$_\bullet$	56.0$_\bullet$	58.7	59.4	60.5	$^\bullet$61.3	$^\bullet$61.5