Effectiveness of Error Correcting Output Coding methods in ensemble and monolithic learning machines

Francesco Masulli[1,2] and *Giorgio Valentini*[2,3]

[1] Dipartimento di Informatica, Università di Pisa
Corso Italia 40, 56125, Pisa, Italy
e-mail masulli@di.unipi.it

[2] INFM - Istituto Nazionale per la Fisica della Materia,
via Dodecaneso 33, 16146 Genova, Italy.

[3] DISI - Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova, via Dodecaneso 35, 16146 Genova, Italy
e-mail: valenti@disi.unige.it,

Corresponding author:
Francesco Masulli
DISI Dipartimento di Informatica
e Scienze dell'Informazione - Università di Genova
Via Dodecaneso 35, 16146 Genova - Italy
Tel: +39 010 353 6604
Fax: +39 010 353 6699
email: masulli@disi.unige.it

Effectiveness of Error Correcting Output Coding methods in ensemble and monolithic learning machines

Abstract

Error Correcting Output Coding (ECOC) methods for multiclass classification present several open problems ranging from the trade-off between their error recovering capabilities and the learnability of the induced dichotomies to the selection of proper base learners and to the design of well-separated codes for a given multiclass problem. We experimentally analyze some of the main factors affecting the effectiveness of ECOC methods. We show that the architecture of ECOC learning machines influences the accuracy of the ECOC classifier, highlighting that ensembles of parallel and independent dichotomic Multi-Layer Perceptrons are well-suited to implement ECOC methods. We quantitatively evaluate the dependence among codeword bit errors using mutual information based measures, experimentally showing that a low dependence enhances the generalization capabilities of ECOC. Moreover we show that the proper selection of the base learner and the decoding function of the reconstruction stage significantly affects the performance of the ECOC ensemble. The analysis of the relationships between the error recovering power, the accuracy of the base learners, and the dependence among codeword bits show that all these factors concur to the effectiveness of ECOC methods in a not straightforward way, very likely dependent on the distribution and complexity of the data.

Keywords: Error Correcting Output Coding; ECOC ensemble; ensemble of learning machines; classification problems.

1 Introduction

In the last decade several methods for constructing ensembles of learning machines have been developed [15]. These methods encompass a wide range of techniques such as ensemble averaging [45, 24], where the outputs of different predictors are linearly combined to produce an overall output; boosting [50, 19, 51] and bagging [9], where the same learning algorithms are applied to different subsets of the training set; mixture of experts [29, 26] where the outputs of the different predictors are non linearly combined through a gating network; ensemble constructed by subsets of input features [10], where each predictor selects a group of the input features.

This paper focuses on Error Correcting Output Coding (ECOC) decomposition methods [16, 17, 32, 31, 28] and in particular on the factors affecting the effectiveness of these ensemble methods.

Error correcting output codes [8], originally proposed to enhance the reliability of the transmission of binary signals through a noisy channel [35], have been successfully used in the framework of decomposition methods for multiclass classification problems to improve the generalization capabilities of learning machines. By this approach an overall classification problem is decomposed into a set of simpler dichotomic subtasks, through a manipulation of the output targets assigned to each class. Dietterich and Bakiri [16, 17] demonstrated that ECOC can achieve better performances than classification methods based on distributed output codes [52]. In fact, using *codewords* for coding the classes suggest the introduction of codes with error recovering abilities. Kong and Dietterich showed that ECOC techniques can also provide class probability informations, through the solution of an over-constrained system of linear equations [31]. An interesting extension of this approach presented by Schapire, consists of the combination of error correcting output codes with boosting techniques [19]; this ensemble method shows good performances on different benchmark machine learning problems [49].

From a statistical standpoint ECOC methods can be viewed as an approximation of a Bayes classifier: James demonstrated that asymptotically, as the the number of dichotomizers approaches infinity, the ECOC classifier will become Bayes consistent (i.e. it always classifies to the Bayes class when the base learner is the Bayes classifier), provided that a random coding matrix is used [27]. In the same perspective Berger showed that randomly selected decomposition matrices are likely to have pairwise well-separate codewords, that is high error recovering capabilities [7]. Variants of the original ECOC algorithms have been proposed, as circular ECOC [21] in order to reduce the sensitivity to codeword selection, or binary labelling techniques [57] to reduce the correlation between the base learners.

The good generalization properties of ECOC methods have been explained through the reduction of both bias and variance [32, 7] and by interpreting them as large margin classifiers [51, 2]. Application of ECOC methods in several domains have shown improvements over standard k-way classification methods. For instance ECOC was successfully applied to classify cloud types [1], for text classification [7, 22], for text-to-speech synthesis [6], to classify olive oils by means of electronic noses [44], for face verification [30] and to classify malignant and normal tissues using gene expression DNA microarray data [54].

ECOC methods present several open problems concerning their properties and the factors affecting their effectiveness. The trade-off between error recovering capabilities and learnability of the dichotomies induced by the decomposition scheme have been tackled in several works [2, 53], but an experimental evaluation of the trade-off has to be performed in order to achieve a better understanding of this phenomenon.

A related problem is the analysis of the relationship between codeword length and performances: some preliminary results seem to show that long codewords improve performance [22].

Another problem, not sufficiently investigated in literature [22, 37, 7], is the proper selection of dichotomic learning machines for the decomposition unit.

Designing codes for a given multiclass problem is another interesting open problem. A greedy approach is proposed in [41], and a method based on soft weight sharing to learn error correcting codes from data is presented in [4]. In [13] it has been shown that given a set of dichotomizers the problem of finding an optimal decomposition matrix is NP-complete: by introducing continuous codes and casting the design problem of continuous codes as a constrained optimization problem, we can achieve an optimal continuous decomposition using standard optimization methods.

In this paper we tackle some of the open problems concerning ECOC methods, and we try to experimentally analyze the factors affecting the effectiveness of ECOC classifiers. In particular we study if the architecture of ECOC learning machines influences the dependence among codeword bit errors and the performance of the overall multiclassifier. Moreover we experimentally compare different decoding functions and different base learners in order to evaluate their influence on the generalization error of the ECOC ensembles. The relationships between ensemble accuracy, base learner accuracy and the error correction power of ECOC codes are experimentally analyzed in order to understand if the error recovering capabilities of ECOC codes can by itself explain the good generalization capabilities of ECOC methods, or if they are the result of complex interactions between the error recovering power of ECOC, the complexity of the induced dichotomies and the accuracy of the base learners composing the ensemble.

In the next section an overview of the ECOC methods is given. Then we experimentally analyze the factors affecting the effectiveness of ECOC methods. A discussion on the results and an outline of future developments of this work concludes the paper.

2 ECOC for multiclass learning problems

In this section we outline the main characteristics of ECOC methods for multiclass classification. ECOC methods code classes through binary strings and exploit the redundancy of the resulting coding schemes to reduce the classification error. They are characterized by a decomposition of a multiclass problem in a set of dichotomic problems and by a successive reconstruction of the original multiclass problem. Two main architectures are feasible, one based on a single learning machine, and another one based on an ensemble of classifiers.

2.1 Decomposition of a multiclass classification problem

In a classification problem based on decomposition methods [36], usually we code classes trough binary strings, or codewords. A coding process is a mapping $\mathcal{M} : \{C_1, \ldots, C_k\} \rightarrow \{s_1, \ldots, s_k\}$ from the set of the classes to the set of binary strings. Each string $s_i, 1 \leq i \leq k$ must univocally determine its corresponding class.

Let be $\mathcal{P} : \mathbf{X} \rightarrow \{C_1, \ldots, C_k\}$ a K classes polychotomy(or *K-polychotomy*), where $\mathbf{X}$ is the multidimensional space of the features and $C_1, \ldots, C_k$ are the labels of the classes. The decomposition of the K-polychotomy generates a set of L dichotomies $f_1, ..., f_L$. Each dichotomy f_i subdivides the input patterns in two complementary superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$, each of them grouping one or more classes of the K-polychotomy. Let be also $D = [d_{ik}]$ a *decomposition matrix* of dimension $L \times K$ that represents the decomposition, connecting classes $C_1, \ldots, C_k$ to the superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$ identified by each dichotomy. An element of D is defined as:

$$d_{ik} = \begin{cases} +1 & \text{if } C_k \subseteq \mathcal{C}_i^+ \\ -1 & \text{if } C_k \subseteq \mathcal{C}_i^- \end{cases} \tag{1}$$

A learning algorithm produces an hypothesis $\hat{\mathbf{f}}(x) = [\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_L(x)]$ of the unknown functional $\mathbf{f}(x) = [f_1(x), f_2(x), \ldots, f_L(x)]$, where $[f_1(x), f_2(x), \ldots, f_L(x)]$ is the codeword s associated by the coding function $\mathcal{M}$ to the class of the input pattern x. When a polychotomy is decomposed into dichotomies, the learning task for $\mathbf{f}(x)$ is reduced to the learning of each $f_i : \mathbf{X} \rightarrow \{-1, 1\}$ through the set of the dichotomizers: it consists in labelling some classes with $+1$ and others with -1. Each dichotomizer is trained to learn f_i, that associates patterns belonging to class C_k with values d_{ik} of the decomposition matrix D, producing the hypothesis $\hat{f}_i$.

In the decomposition matrix, rows correspond to dichotomizers tasks and columns to classes. In this way, each class is univocally determined by its specific codeword. Using ECOC codes as codewords we can achieve a so-called ECOC decomposition. For instance, considering a decomposition matrix for a 4 class classification problem with 7-bit class coding (Tab. 1), the task of the second dichotomizer, namely f_2, consists in separating the patterns belonging to classes C_1 and C_4 from the patterns of class C_2 and C_3. The third column of the decomposition matrix represents the codeword $[-1, -1, +1, +1, -1, -1, +1]$ associated to the class C_3.

ECOC decomposition tries to maximize error recovering capabilities through the maximization of the minimum distance between each couple of codewords [32, 36].

Several methods for generating ECOC codes have been proposed: exhaustive codes, randomized hill climbing [17], Hadamard and BCH codes [8, 46], and random codes [27], but open problems are still the joint maximization of distances between rows and columns of the decomposition matrix.

2.2 Reconstruction and decoding

After the training of the dichotomizers $\hat{f}_i$, their outputs are used to reconstruct the polychotomy in order to determine the class $C_i \in \{C_1, \ldots, C_k\}$ of the input patterns, using a suitable measure of similarity.

Learning machines constructed by ECOC are made up by a *Decomposition Unit* and a *Decision Unit*. The Decomposition Unit analyzes the input patterns and calculates the codewords using an assigned decomposition scheme generated by a suitable algorithm. This unit computes:

$$\hat{\mathbf{f}}(x) = [\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_L(x)] \tag{2}$$

The Decision Unit decodes the computed codeword $\hat{s} = [\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_L(x)]$, mapping it to the associated class. This unit computes the function $\mathcal{D} : S \rightarrow C$:

$$\mathcal{D}(\hat{s}) = \mathcal{D}[\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_L(x)] \tag{3}$$

where S is the set of the computed codewords, C is the set of the classes, $\hat{f}_i(x)$ are the hypotheses returned by the learning algorithm, and $\mathcal{D}$ is a suitable decoding function.

The Decision Unit decodes the codeword computed by the Decomposition Unit, choosing the class whose codeword is more *similar* to that computed by the set of dichotomizers. So, the decoding function $\mathcal{D}(\hat{s})$ can be implemented by a maximization of a similarity measure between the computed $\hat{s}$ codeword and the effective codewords $s_i, 1 \leq i \leq K$ associated to the classes:

$$class_{out} = \mathcal{D}(\hat{s}) = \arg \max_{1 \leq i \leq K} Sim(\hat{s}, s_i) \tag{4}$$

where $class_{out}$ is the class computed by the polychotomizer, s_i is the codeword of class C_i, the vector $\hat{s}$ is the codeword computed by the set of dichotomizers, and $Sim(x, y)$ is a general similarity measure between two vectors x and y. This similarity measure can be a Hamming distance if the outputs of the dichotomizers $\hat{s} = [\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_L(x)]$ are discrete or the inner product, or L_1 or L_2 norm distances for dichotomizers with continuous outputs. For instance, using the inner product we have:

$$\mathcal{D}_{prod}(\hat{s}) = \arg \max_{1 \le i \le K} (\hat{s} \dot{s}_i) \qquad (5)$$

Dietterich and Bakiri [16, 17] proposed the *Error Correcting Output Coding* (ECOC) decomposition scheme with the aim of improving the generalization capabilities of NETtalk classifier systems based on distributed output codes [52]: Coding the classes by codewords suggests the idea of adding *error recovering* capabilities to decomposition methods in order to obtain classifiers less sensitive to noise [32, 37]. This goal is achieved by means of the redundancy of the coding scheme, as shown by coding theory [56].

The error–recovering capabilities of ECOC codes depend mainly on column separation, i.e. the distance between codewords must be increased, according to an assigned measure. The maximal number of errors Max_{err} that can be corrected in an ECOC based decomposition is [32]:

$$\text{Max}_{err} = \left\lfloor \frac{\Delta_D - 1}{2} \right\rfloor \qquad (6)$$

where Δ_D is the minimal Hamming distance between each pair of columns of the decomposition matrix D.

2.3 Design of ECOC classifiers

There are two main approaches to the design of a classifier using ECOC codes, based on the features of the Decomposition Unit. The first approach, that we call *Monolithic Classifiers*, makes use of a Decomposition Unit based on a multiple inputs multiple outputs (MIMO) learning machine exploiting the decomposition in an implicit way. *Monolithic Classifiers* are,e.g., MIMO MLPs or MIMO decision trees [16, 17].

The second approach, that we call *Parallel Classifiers*, makes use of a Decomposition Unit leading to the distribution of the learning task among separated and independent dichotomizers that can be implemented through, e.g., Support Vector Machines [12], multiple inputs single output (MISO) MLP or dichotomic decision trees. We call the resulting learning machines *Parallel Linear Dichotomizers* (*PLD*) if the dichotomizers used for implementing the dichotomizers are linear (as in [3]), or *Parallel Non-linear Dichotomizers* (*PND*) if the dichotomizers are non-linear [17, 38].

Parallel Non-linear Dichotomizers (*PND*) are multiclassifiers based on the decomposition of polychotomies into dichotomies, using dichotomizers solving their classification tasks independently from each other [38]. In the decomposition unit each dichotomizer is implemented by a separate *non-linear* learning machine, and learns a different and specific dichotomic task using a training set common to all the dichotomizers. The decision unit can use a L_1 norm or another similarity measure between codewords to predict classes of unlabeled patterns. *PND* have been implemented with decision trees [17, 32, 49], or dichotomic MLP [38].

Parallel Linear Dichotomizers (*PLD*) are also multiclassifiers based on decomposition of polychotomies into dichotomies, but each dichotomizer is implemented by a separate linear learning machine [3].

It is worth noting that classifiers based on decomposition methods and classifiers based on ensemble averaging methods [45, 24] or bagging and boosting [9, 19] share the idea of using a set of learning machines acting on the same input and recombining their outputs in order to make decisions; the main difference lies in the fact that in classifiers based on decomposition methods the task of each learning machine is specific and different from that of the others.

3 Effectiveness of ECOC methods

In this section we experimentally analyze the factors affecting the effectiveness of ECOC methods. In particular we focus on the following items:

1. Architecture of the decomposition unit.
2. Dependency among codeword bits coding the classes.
3. Decoding function selected for the decision unit.
4. Relationships between ensemble accuracy, base learner accuracy and error correcting power.

In the following sections we address each problem separately.

3.1 Architectures

In this section we study two different architectures for ECOC learning machines, considering decomposition units composed by a single *monolithic* MLP, that learns the codewords as a whole, and by an ensemble of dichotomic MLP, each learning a different bit of the codewords coding the classes.

We apply also the widely used One-Per-Class [1] (OPC) [43, 5] decomposition scheme as reference comparison, both using MLP monolithic and MLP parallel classifiers decomposition units.

[1] In a One-Per-Class (OPC) decomposition scheme, each dichotomizer f_i have to separate a single class from all the others.

PND are implemented by a set of multi-layer perceptrons with a single hidden layer, acting as dichotomizers, and *PLD* are implemented by a set of single layer perceptrons.

Monolithic MLP are built using a single hidden layer and sigmoidal activation functions, both in hidden and output neurons. The number of neurons of the hidden layer amounts from ten to one hundred according to the complexity of the data set to be learned. The base learners of the *PND* and the monolithic MLP have been trained using the backpropagation algorithm with fixed learning rate.

We have compared the performances of the three ECOC learning machines using different data sets, both real and synthetic, as shown in Tab. 2. The data sets *p6* and *p9* (available by anonymous ftp at `ftp://ftp.disi.unige.it/person/ValentiniG/Data`), are synthetic and composed by normal distributed clusters of data. The set *p6* contains 6 classes with no overlapping regions, while the regions of the 9 classes of *p9* hardly overlap. *Glass*, *letter* and *optdigits* data sets are from the *UCI repository* [42].

In the experimentation we have used exhaustive [17] and BCH ECOC generation algorithms [8]. ECOC exhaustive algorithms select among all possible 2^K dichotomies the $2^{K-1} - 1$ ones that are not equivalent and not trivial [2]. ECOC obtained by exhaustive algorithms are Bayes consistent [27], i.e. if the component dichotomizers approximate the Bayes optimal discriminant function, then the overall polychotomizer will produce an optimal Bayes classification. The shortcoming of the algorithm is the exponential growth

[2] Dichotomies f' and f such that $f' = f$ or $f' = -f$ are equivalent, while for trivial dichotomies $f^{-1}(+1) = \emptyset$ or $f^{-1}(-1) = \emptyset$ holds.

of the codeword lengths with the cardinality of the classes. BCH ECOC are computed using an algebraic method based on a polynomial representation of finite Galois fields [8]. BCH ECOC are not Bayes optimal, but allow to generate ECOC codewords of tractable length. We have modified the original algorithm removing duplicate rows of the decomposition matrix or trivial dichotomies, and deleting rows with a Hamming distance equal or less than an assigned threshold. This modified version of the algorithm has produced 7 bits ECOC codes for the data sets *p6* and *glass*, 15 bits codes for *p9* and *optdigits* and 30 bit ECOC codes for *letter* data set.

The programs used in our experiments have been developed using *NEURObjects* [55], a C++ library for neural networks development. In the experimentation we used resampling methods, using a single pair of training and testing data set or *k-fold cross validation*, to estimate the generalization error of ECOC monolithic and ECOC *PND* ensembles.

Fig. 2, shows the performances of MLP, PLD and PND over the considered data sets. ECOC MLP monolithic classifiers do not outperform standard MLP (Fig. 2 (a)). In [17] similar results have been obtained over the same data set *letter* we used.

Concerning *PLD* (Fig. 2 (b)), over data sets *p6*, *p9*, and *optdigits* there is no significant statistical difference among OPC and ECOC decomposition, while over *glass PLD* ECOC outperforms all other types of polychotomizers, but with *letter PLD* OPC achieve better results.

Considering *PND* (Fig. 2 (c)), for data sets *p6* and *optdigits* no significant differences among OPC and ECOC *PND* can be noticed. Over the *p9* data set, ECOC shows expected errors significantly smaller than OPC.

Expected errors over *glass* and *letter* data sets are significantly smaller for ECOC compared with OPC. So we can see that ECOC *PND* show expected error rates significantly lower than OPC *PND*.

The differences in performances between OPC and ECOC decomposition schemes are considered statistically significant if their confidence level is less or equal to 0.05 according to *McNemar* test [18] or *k fold cross validated paired t test* [14].

PLD show remarkable higher errors over all data sets, and in particular they fail over *p9*. Summarizing, the expected errors are significantly smaller for *PND* compared with direct monolithic MLP classifiers and *PLD*, and ECOC outperforms OPC decomposition only in *PND* ensembles.

3.2 Dependency among codeword bits

Several authors noted that the dependence among output errors affect the effectiveness of ECOC methods [32, 23, 37], and recently we provided a quantitative evidence of this fact [40]. In the domain of the serial transmission of messages coded as sequences of bits, Peterson and Weldon [46] showed that if errors on different codeword bits are dependent, the effectiveness of error correcting code is reduced. Transferring these outcomes in the framework of classification problems, if a decomposition matrix contains very similar rows (dichotomies), each error of an assigned dichotomizer will be likely to appear in the most correlated dichotomizers, thus reducing the effectiveness of ECOC.

In order to quantitatively evaluating the dependence among output errors of the decomposition unit of ECOC learning machines, we used mutual information based measures proposed in [39].

Mutual information, being a special case of the Kullback-Leibler divergence between two distributions, measures the matching between the joint probability density distribution and the product of the marginal probability density distribution of the output errors. If we have a complete matching, the mutual information is 0 and the output errors are independent, otherwise the higher the value of the mutual information between output errors is, the higher the dependence between them will be.

Fig. 3 shows the compared *mutual information error index* $\varPhi_R$ [39] between *monolithic* and *PND* ECOC learning machines considering 4 different data sets. It is defined as:

$$\varPhi_R = \sum_{i=1}^{l} \sum_{j=1}^{l} I_E(e_i, e_j) \tag{7}$$

where $I_E(e_i, e_j)$ is the mutual information between the errors of the i^{th} and j^{th} output of the decomposition unit.

On the axes are represented the $\varPhi_R$ values of ECOC *monolithic* and ECOC *PND* learning machines. This index measures the sum of the mutual information of the output errors between all the output pairs of the learning machines, giving a computable quantity to estimate the dependence between codeword bit errors: an high value of $\varPhi_R$ corresponds to an high dependence between output errors and vice versa. Each point corresponds to ECOC learning machines implemented with MLP with different number of hidden units and using different partitions of the output error. On all the data sets about all the points are above the dotted line, i.e. all the values of $\varPhi_R$ are greater for ECOC *monolithic* compared with ECOC *PND*. The results show that *monolithic* architectures are affected by a higher dependence among codeword bit errors, confirming the results obtained in [40].

3.3 Relationships between ensemble accuracy and decoding function

In this section we experimentally study if the choice of a particular decoding function affects the performance of ECOC MLP ensembles. We analyze also the effectiveness of the ECOC ensemble varying the minimum Hamming distance with fixed length codewords, as the error recovering capabilities of the ECOC ensembles depend critically on the minimum Hamming distance between codewords (eq. 6).

The decoding in Output Coding methods is performed using similarity measures between the computed codeword and the codeword coding the classes (Sect. 2). We consider three commonly used decoding functions based on Hamming distance and L_1 and L_2 norm distance.

Given a $n \times k$ decomposition matrix D (eq. 1), we indicate with D_{ij} the i^{th} bit of the j^{th} codeword coding the classes, and with $C_i(x)$ the output computed by the i^{th} dichotomizer on the input $x \in \mathbb{R}^d$. If $C_i(x) \in \{-1, +1\}$, then the decoding function based on the Hamming distance is:

$$\mathcal{D}_{Hamm}(x) = \arg\min_j \sum_{i=1}^{n} \frac{1}{2}|D_{ij} - C_i(x)| \tag{8}$$

If $C_i(x) \in \mathbb{R}^d$, then the decoding function based on the L_1 norm distance is:

$$\mathcal{D}_{L_1}(x) = \arg\min_j \sum_{i=1}^{n} |D_{ij} - C_i(x)| \tag{9}$$

and the decoding function based on the L_2 norm distance is:

$$\mathcal{D}_{L_2}(x) = \arg\min_j \sum_{i=1}^{n} (D_{ij} - C_i(x))^2 \tag{10}$$

We generated ECOC decomposition matrices with constrained random algorithms. The random generation have been constrained in order to eliminate trivial dichotomies (e.g. rows with all +1 or all −1), and equal or complementary rows to assure the absence of equal and equivalent symmetric dichotomies, and to achieve a desired minimum Hamming distance between the columns (codewords) of the decomposition matrix [3].

The generalization error of the ECOC ensembles have been estimated using 5-fold cross validation. We merged the training and test sets of the optdigits and image-segmentation data sets from the UCI repository to perform a 5-fold cross validation on the overall merged data set. We used also p20, a synthetic data set generated through the *NEURObjects* application `dodata`. It is composed by 20 3-dimensional classes, and each class is characterized by 3 disjoint clusters of data normally distributed with diagonal covariance matrices [4].

The comparison of the estimated generalization error of the ECOC ensembles shows that the decoding functions based on the L_1 and L_2 norm distances outperform the decoding based on the Hamming distance (Fig 4, 5 and 6). Only on the *p20* data set with 50-bit ECOC ensembles with linear perceptrons as base learners there is no difference between L_1 norm and Hamming distance based decoding, but in this task the ECOC ensemble clearly

[3] the C++ classes implementing the ECOC random algorithms are available at `http:/ftp.disi.unige.it/person/ValentiniG/SW/ECOCrandom`

[4] p20 is available at `http:/ftp.disi.unige.it/person/ValentiniG/Data/p20.tgz`

fails, performing a sort of random guessing. The same figures show also that there is no significant difference between L_1 and L_2 norm distance decoding, considering the proposed data sets.

Considering the relationship between the estimated generalization error and the minimum Hamming distance (MHD) between the codewords, at first glance, we could expect a monotonic decrement with the MHD. But the trends seem to be more complex: they are relatively irregular (Fig 4, 5 and 6) and only in a few cases (Fig. 4 c, d) we can observe a monotonic decrement of the error with MHD.

3.4 Relationships between ensemble and base learner accuracy and error correcting power

The previous section showed that the increment of the MHD is not by itself sufficient to reduce the error of the ensembles. In this section we study how the characteristics of the base learner and in particular its accuracy affects the overall performance of the ECOC ensemble.

As expected, the selection of the base learner affects the overall performance of the ensemble (Fig 7). Indeed, using MLP base learner with different number of hidden units greatly influences the estimated generalization error of the ensemble. For instance, on the *p20* data set, ECOC ensembles having as base learners MLP with 24 hidden units are able to halve the estimated generalization error of ECOC ensembles with MLP with 6 hidden units (Fig. 7 (d)). Also on the optdigits and image-segmentation data set (Fig. 7 a,b and c) the selection of different MLP as base learner has a significant impact on the performance of the ensemble.

In order to investigate why the selection of different base learners affects in a so significant manner the performance of the ensemble, we analyzed the relationship between the overall ensemble error, the average base learner error and the minimum Hamming distance between the codewords.

With the *optdigits* data set the average base learner error increments with the minimum Hamming distance (Fig. 8), while the ensemble error tends to decrease with the MHD, especially using long codewords (Fig. 8 c, d and f). More irregular patterns are observed in the *image-segmentation* data set: the average base learner error increases with the Hamming distance, but in an irregular way and the ensemble error rate oscillates around 0.04 and 0.035 (Fig. 9). With the *p20* synthetic data set the base learner average error initially grows, then goes down and hence increments with the MHD, while the ensemble error decreases only using base learners with 6 hidden units: using more complex base learners the ensemble error shows an irregular trend, with minima of the error for 11 and 17 bit MHD (Fig. 10). Note that the ensemble error partially follows the average base learner error, with a decrement for increasing MHD values, but sometimes we can also observe opposite trends of the average base learner and ensemble errors (Fig. 8, 9 and 10).

The graphs of Fig. 11 represent directly the ensemble error in function of the the average base learner error. Each point in the graph corresponds to a different MHD between codewords. Considering different data sets and different base learners we observe very different trends: with the *optdigits* data set and MLP base learners we have a decrement of the ensemble error even if the average base learner error increases (Fig. 11 a), but with the same data set using linear perceptrons base learners we have an increment of the error, especially with 32-bit codewords (Fig. 11 b). With the *p20* and *image-segmentation* data sets the trends seem to be more complex, without a clear relationship between ensemble and average base learner accuracy (Fig. 11 c and d). Summarizing we cannot observe simple relationships between the ensemble error and the average base learner error with respect to the MHD between codewords.

4 Discussion

The experimental results of the previous section show that a set of different factors affect the effectiveness of the ECOC methods. The architecture of the ECOC learning machine, the dependency among codeword bits, the type of the decoding function, the error recovering power of the decomposition scheme, and the accuracy of the dichotomizer interact between them and contribute to the effectiveness of the ECOC methods.

The design of the decomposition unit of the ECOC learning machine affects the performances, as shown by our experimentation. In [16, 32] it has been stated that ECOC classifiers should be preferred to OPC classifiers, as they reduce error bias and variance more than standard classifiers; their experimental results confirm these hypotheses, with the exception of some cases over complex data sets (such us *letter* from UCI repository) where OPC MLP classifiers perform better than ECOC MLP. Our experimentation has pointed out that not always ECOC monolithic MLP outperform OPC MLP classifiers, while we have found a significant difference between ECOC and OPC *PND* performances (Fig. 2). We suppose that the better performances of *PND* can be explained considering on one hand that their dichotomizers are less complex than ECOC *monolithic* learning machines, achieving by this way better generalization capabilities. On the other hand, ECOC *monolithic* learning machines introduce more correlation among codeword bits. In fact in *PND* each codeword bit is learned and computed by its own MLP, specialized for its particular dichotomy, while in monolithic classifiers each codeword bit is learned and computed by a linear combination of hidden layer outputs pertaining to one and only shared multi-layer perceptron. Concerning PLD, we point out that the error recovering capabilities induced by ECOC are counter-balanced by higher error rates of linear dichotomizers (Fig. 5 d, and Fig. 6 d). Hence, interdependence among monolithic MLP ECOC outputs lowers the effectiveness of ECOC codes for this kind of classifiers. This is conformed also by the quantitative evaluation of the dependence among codeword bit errors we performed using mutual information based measures: ECOC *PND* ensembles show a lower dependence compared with monolithic ECOC MLP.

The results showed also that the decoding function plays an important role: indeed L_1 and L_2 norm distance seem to be well-suited for the decoding, while the Hamming distance based decoding function achieves worse results. This is not surprising, as L_1 and L_2 norms exploit the "confidence" in the prevision of each base learner, while Hamming decoding discards all the information except the hard membership to a class.

The choice of proper dichotomizers, well-suited for a given decomposition, affects also the performances of ECOC ensembles, as shown by our experimental results (Fig. 7): in general complex classification problems need more complex dichotomizers, but overfitting phenomena can also arise.

The error recovering power of ECOC methods depends on the MHD between codewords (if the output errors of the decomposition unit are independent). Our experiments show that if we use fixed length codewords an increment of the MHD does not lead necessarily to improved performances of the ECOC ensemble. This can be explained considering that different codewords induce different dichotomies. The dichotomies can or cannot be hard learnable depending on the structure of the data and on the type of the base learner used. The learnability is partially reflected by the average base learner error. As shown by our experiments there is not a simple relationship between ensemble error, average base learner error and the MHD. In some cases the effect due to the error recovering power prevails on the increment of the base learner error (Fig. 8 c and d); in other cases the error recovering power is counter-balanced by the increased average base learner error (Fig. 8 f); in other cases we can also have similar trends of the ensemble error and the average base learner error with respect to MHD (Fig. 9 b and c). We can also observe in a few cases that the average base learner error decreases and the ensemble error increases for some values of the MHD (Fig. 9 a and c). This seems to be counter-intuitive, but we know that the average base learner error does not take into account the distribution of the error among the dichotomies: we can have very different distributions of the error (and very different ensemble errors) with the same average base learner error. Tab. 3 shows that there is a relatively large variability of the base learner error. Hence our results show that there is not a simple relationship between error recovering power and accuracy of the induced dichotomies with respect to the accuracy of the ensemble. It is not sufficient to increase the MHD to improve the performance: we have to take into account the accuracy of the dichotomizers and the dependency among the codeword bit errors.

Following the approach of Ghani [22], with some strong assumptions, we can try to model ECOC ensemble error through the binomial distribution. In fact if we assume that the probability of error of each dichotomizer has about the same value p, and assuming that the outcomes of the n dichotomizers of the ECOC ensemble are independent, the probability of k errors on n trials (dichotomizer tasks) with equal probability of error p is distributed according to a binomial distribution:

$$P(k, n, p) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}$$

Considering that ECOC codes with $MHD = min$ can correct $(min - 1)/n$ errors (eq. 6), then the overall probability of error of an ECOC ensemble with average base learner error p, with $kmin = \frac{min-1}{n} + 1$, is given by:

$$P_{err} = \sum_{k=kmin}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} \tag{11}$$

Tab. 4 compares the experimental error of the ECOC ensemble and the theoretical error estimated through the binomial distribution (eq. 11). The results show that only in some cases the experimental and the theoretical error agree: especially for relatively low Hamming distance between codeword the difference is very significant. This is not too surprising, because our assumption about the independence of codeword bit errors and the equal probability of error for all the dichotomizers are too strong. In fact our previous results (Sect. 3.2) showed that a correlation between codeword bit errors does exist, depending on the type of the decomposition, the base learner used and the structure of the data, and also the probability of errors of the different dichotomizers show a not negligible variance (Tab. 3). These assumptions can or cannot hold depending on the structure of the data, the type of decomposition and the base learner used.

An important item only indirectly studied in this work is the relationship between the error recovering power and the complexity of the induced dichotomies. Although the average base leaner error can be informative about the complexity of the dichotomies, this information is biased by the characteristics of the base learner: the functions implemented by a particular base learner can perform better on certain data distributions, but can can achieve worse performances on others [11].

We need more studies to relate the accuracy of the ECOC ensemble with the complexity of the induced decomposition. The relationship effectiveness of ECOC ensemble-complexity of the data is an item common to other ensemble methods [25] and require specific studies and experimental analysis using appropriate measures of complexity, based on geometrical or topological characteristics of data [34].

5 Conclusions

The effectiveness of ECOC methods depends on many factors ranging from the architecture of the decomposition unit, to the dependence among codeword bits coding the classes, the decoding function selected for the decision unit, the error recovering power of the ECOC codes, the type and accuracy of the base learners of the ensemble, the complexity of the dichotomies induced by the ECOC decomposition.

The results of our experiments suggest that ensembles of learning machines achieve in general better results than single monolithic learning machines, as dedicated and independent base learners reduce the correlation among the codeword bits, and their learning tasks are reduced to dichotomies in general simpler than polychotomies.

The dependence among codeword bits reduces the error recovering power of ECOC: improving the diversity of the dichotomies and of the dichotomizers can enhance the performance of ECOC learning machines.

Using Minkowski norm in decoding functions of the decision unit seems to be more reliable and robust than using the Hamming distance.

The selection of proper base learners influences the accuracy of the ensemble: a possible way of research could be to experiment with different and specific dichotomic learning machines well-suited for each different dichotomic problem induced by the decomposition.

ECOC codes can recover errors committed by the base learners, but increasing the minimum Hamming distance between codewords, does not lead by itself to better performances, because we could have a contemporary increased complexity of the induced dichotomies, or more similar and correlated dichotomies.

Increasing the codeword length, as shown in [22, 7] can in general lead to better performances, but for fixed-length codewords our experimentation showed that many factors interact to determine the effectiveness of ECOC methods. We know that the problem of finding an optimal decomposition matrix is NP-complete [13], and our experimental analysis showed that no straightforward solution exists to select jointly low-correlated dichotomies, codewords with high minimum Hamming distance and simple induced dichotomies, and base learner well-suited for a given decomposition. The main problem arising from our experimental analysis consist in evaluating how the complexity of the data characterizing a given classification problem affects the performance of ECOC methods with respect to the error recovering power of the ECOC decomposition and the accuracy of the base learner of the ensemble. This in turn requires the definition and usage of proper data complexity measures, such as the length of class boundary [20] or the ϵ-neighborhoods space covering [33] to characterize the classification problem complexity on the basis of the geometrical and topological properties of the classes.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. This work was supported by INFM, University of Genova, and Madess II CNR.

References

[1] D. Aha and R. Bankert. Cloud classification using error-correcting output codes. In *Artificial Intelligence Applications: Natural Science, Agriculture and Environmental Science*, volume 11, pages 13–28. 1997.

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

[3] E. Alpaydin and E. Mayoraz. Combining linear dichotomizers to construct nonlinear polychotomizers.PAAreview Technical Report IDIAP-RR 98-05, IDIAP - Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, 1998.

[4] E. Alpaydin and E. Mayoraz. Learning error-correcting output codes from data. In *ICANN'99*, pages 743–748, Edinburgh, UK, 1999.

[5] R. Anand, G. Mehrotra, C.K. Mohan, and S. Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6:117–124, 1995.

[6] G. Bakiri and T.G. Dietterich. Achieving high accuracy text-to-speech with machine learning. In *Data mining in speech synthesis*. 1999.

[7] A. Berger. Error correcting output coding for text classification. In *IJCAI'99: Workshop on machine learning for information filtering*, 1999.

[8] R.C. Bose and D.K. Ray-Chauduri. On a class of error correcting binary group codes. *Information and Control*, (3):68–79, 1960.

[9] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[10] K.J. Cherkauker. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In

Chan P., editor, *Working notes of the AAAI Workshop on Integrating Multiple Learned Models*, pages 15–21. 1996.

[11] S. Cohen and N. Intrator. Automatic Model Selection in a Hybrid Perceptron/Radial Network. In *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 349–358. Springer-Verlag, 2001.

[12] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[13] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 35–46, 2000.

[14] T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 1998.

[15] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2000.

[16] T.G. Dietterich and G. Bakiri. Error - correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of AAAI-91*, pages 572–577. AAAI Press / MIT Press, 1991.

[17] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.

[18] B.S. Everitt. *The analysis of contingency tables.* Chapman and Hall, London, 1977.

[19] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kauffman, 1996.

[20] J.H. Friedman and L.C. Rafsky. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistics*, 7(4):697–717, 1979.

[21] R. Ghaderi and T. Windeatt. Circular ECOC, a theoretical and experimental analysis. In *Int. Conf. Pattern Recognition, Barcelona, Spain*, pages 203–206, 2000.

[22] R. Ghani. Using error correcting output codes for text classification. In *ICML 2000: Proceedings of the 17th International Conference on Machine Learning*, pages 303–310, San Francisco, US, 2000. Morgan Kaufmann Publishers.

[23] V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In *Proc. of the Twelfth Annual Conference on Computational Learning Theory*, pages 145–155. ACM Press, 1999.

[24] S. Hashem. Optimal linear combinations of neural networks. *Neural Computation*, 10:599–614, 1997.

[25] T.K. Ho. Data Complexity Analysis for Classifiers Combination. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second Inter-*

national Workshop, MCS 2001, Cambridge, UK, volume 2096 of *Lecture Notes in Computer Science*, pages 53–67, Berlin, 2001. Springer-Verlag.

[26] R.A. Jacobs. Methods for combining experts probability assessment. *Neural Computation*, 7:867–888, 1995.

[27] G. James. *Majority vote classifiers: theory and applications.* PhD thesis, Department of Statistics - Stanford University, Stanford, CA, 1998.

[28] G. James and T. Hastie. The error coding method and pict. *J. of computational and graphical statistics*, 7(3):377–387, 1998.

[29] M.I. Jordan and R.A. Jacobs. Hierarchical mixture of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[30] J. Kittler, R. Ghaderi, T. Windeatt, and G. Matas. Face Verification using Error Correcting Output Codes. In *Computer Vision and Pattern Recognition CVPR01, Hawaii, USA*, pages 755–760, 2001.

[31] B.E. Kong and T.G. Dietterich. Probability estimation via error correcting output coding. In *IASTED International Conference: Artificial Intelligence and Soft Computing*, Banff, Canada.

[32] E. Kong and T.G. Dietterich. Error - correcting output coding correct bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kauffman.

[33] F. Lebourgeois and H. Emptoz. Pretopological approach for supervised learning. In 13^{th} *Int. Conf. Pattern Recognition, Wien, Austria*, pages 256–260, 1996.

[34] M. Li and P Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer-Verlag, Berlin, 1993.

[35] S. Lin and D.J.Jr. Costello. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Englewood Cliffs, 1983.

[36] F. Masulli and G. Valentini. Comparing decomposition methods for classification. In R.J. Howlett and L.C. Jain, editors, *KES'2000, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pages 788–791, Piscataway, NJ, 2000. IEEE.

[37] F. Masulli and G. Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In *Lecture Notes in Computer Science*, volume 1857, pages 107–116. Springer-Verlag, Berlin, Heidelberg, 2000.

[38] F. Masulli and G. Valentini. Parallel Non linear Dichotomizers. In *IJCNN2000, The IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 2, pages 29–33, Como, Italy, 2000.

[39] F. Masulli and G. Valentini. Mutual information methods for evaluating dependence among outputs in learning machines. Technical Report TR-01-02, DISI - Dipartimento di Informatica e Scienze dell' Informazione - Università di Genova, 2001. ftp://ftp.disi.unige.it/person/ValentiniG/papers/TR-01-02.ps.gz.

[40] F. Masulli and G. Valentini. Quantitative Evaluation of Dependence among Outputs in ECOC Classifiers Using Mutual Information Based

Measures. In K. Marko and P. Webos, editors, *Proceedings of the International Joint Conference on Neural Networks IJCNN'01*, volume 2, pages 784–789, Piscataway, NJ, USA, 2001. IEEE.

[41] E. Mayoraz and M. Moreira. On the decomposition of polychotomies into dichotomies. In *The XIV International Conference on Machine Learning*, pages 219–226, Nashville, TN, July 1997.

[42] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases, 1998. www.ics.uci.edu/mlearn/MLRepository.html.

[43] N.J. Nilsson. *Learning Machines.* Mc Graw Hill, New York, 1965.

[44] M. Pardo, G. Sberveglieri, F. Masulli, and G. Valentini. Decompositive classification models for electronic noses. *Anal. Chimica Acta*, 446:223–232, 2001.

[45] M.P. Perrone and L.N. Cooper. When networks disagree: ensemble methods for hybrid neural networks. In Mammone R.J., editor, *Artificial Neural Networks for Speech and Vision*, pages 126–142. Chapman & Hall, London, 1993.

[46] W.W. Peterson and E.J.Jr. Weldon. *Error correcting codes.* MIT Press, Cambridge, MA, 1972.

[47] J.R. Quinlan. *C4.5 Programs for Machine Learning.* Morgan Kauffman, 1993.

[48] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In Rumelhart D.E., McClelland J.L.,

editor, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8. MIT Press, Cambridge, MA, 1986.

[49] R.E. Schapire. Using output codes to boost multiclass learning problems. In *Proceedings of the Fourteenth International Conference on on Machine Learning*, pages 313–321, San Francisco, CA, 1997. Morgan Kauffman.

[50] R.E. Schapire. A brief introduction to boosting. In Thomas Dean, editor, *16th International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kauffman, 1999.

[51] R.E. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[52] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Journal of Artificial Intelligence Research*, (1):145–168, 1987.

[53] G. Valentini. Upper bounds on the training error of ECOC-SVM ensembles. Technical Report TR-00-17, DISI - Dipartimento di Informatica e Scienze dell' Informazione - Università di Genova, 2000. ftp://ftp.disi.unige.it/person/ValentiniG/papers/TR-00-17.ps.gz.

[54] G. Valentini. Classification of human malignancies by machine learning methods using DNA microarray gene expression data. In G.M. Papadourakis, editor, *Fourth International Conference Neural Networks*

and Expert Systems in Medicine and HealthCare, pages 399–408, Milos island, Greece, 2001.

[55] G. Valentini and F. Masulli. NEURObjects: an object-oriented library for neural network development. *Neurocomputing.* (in press).

[56] J. Van Lint. *Coding theory.* Spriger Verlag, Berlin, 1971.

[57] T. Windeatt and R. Ghaderi. Binary labelling and Decision Level Fusion. *Information Fusion*, 2(2):103–112, 2001.

List of Tables

1 ECOC decomposition matrix example. 36

2 Data sets general features. The *glass*, *letter* and *optdigits* data sets are from the *UCI repository* [42]. 37

3 Average base learner error and its variation. The last two columns refer to the minimum and average diversity of the dichotomies. MHD stands for Minimum Hamming Distance, Stdev for standard deviation, MRD for Minimum Row Distance and ARD for Average Row Distance 38

4 Experimental ensemble error and theoretical ensemble error predicted through the binomial distribution. 39

Table 1: ECOC decomposition matrix example.

Dichotomizers tasks	Columns: class codewords			
	C_1	C_2	C_3	C_4
f_1	+1	-1	-1	-1
f_2	+1	-1	-1	+1
f_3	+1	-1	+1	-1
f_4	+1	-1	+1	+1
f_5	+1	+1	-1	-1
f_6	+1	+1	-1	+1
f_7	+1	+1	+1	-1

Table 2: Data sets general features. The *glass*, *letter* and *optdigits* data sets are from the *UCI repository* [42].

Data set	Number of attributes	Number of classes	Number of training samples	Number of testing samples
*p*6	3	6	1200	1200
*p*9	5	9	1800	5-fold cross-val
glass	9	6	214	10-fold cross-val
letter	16	26	16000	4000
optdigits	64	10	3823	1797

Table 3: Average base learner error and its variation. The last two columns refer to the minimum and average diversity of the dichotomies. MHD stands for Minimum Hamming Distance, Stdev for standard deviation, MRD for Minimum Row Distance and ARD for Average Row Distance

Data set	MHD	Average base learner err.	Stdev base learner err.	Min. base learner err.	Max. base learner err.	MRD	ARD
p20 50-bit ECOC	9	0.0912	0.0425	0.0358	0.2675	2	10.075
	11	0.0846	0.0299	0.0358	0.1641	4	9.984
	13	0.0831	0.0260	0.0325	0.1483	4	9.969
	15	0.0816	0.0327	0.0291	0.2150	3	10.030
	17	0.0801	0.0276	0.0325	0.1650	1	10.048
	18	0.0857	0.0255	0.0300	0.1366	3	9.958
	19	0.0939	0.0323	0.0358	0.1908	3	10.018
	20	0.0942	0.0321	0.0358	0.1908	3	9.994
optdigits 64 bit ECOC	15	0.1094	0.1292	0.0042	0.5592	1	4.979
	17	0.1125	0.1296	0.0071	0.5368	1	4.987
	19	0.1336	0.1381	0.0133	0.5386	1	4.959
	21	0.1332	0.1512	0.0071	0.5699	1	4.997
	23	0.1220	0.1281	0.0142	0.5686	1	5.038
	25	0.1473	0.1566	0.0138	0.5919	1	4.991
	27	0.1476	0.1543	0.0090	0.5238	1	4.968
	28	0.1514	0.1557	0.0138	0.5339	1	5.025

Table 4: Experimental ensemble error and theoretical ensemble error predicted through the binomial distribution.

Data set	Min. Hamm. dist.	Average base learner err.	Experimental err.	Theoretical err.
p20 50-bit ECOC	9	0.0912	0.0808	0.4846
	11	0.0846	0.0708	0.2462
	13	0.0831	0.0800	0.1187
	15	0.0816	0.0766	0.0483
	17	0.0801	0.0633	0.0167
	18	0.0857	0.0741	0.0249
	19	0.0939	0.0833	0.0165
	20	0.0942	0.0816	0.0168
optdigits 64 bit ECOC	15	0.1094	0.0674	0.4022
	17	0.1125	0.0695	0.2910
	19	0.1336	0.0750	0.3493
	21	0.1332	0.0651	0.2278
	23	0.1220	0.0663	0.0847
	25	0.1473	0.0725	0.1401
	27	0.1476	0.0660	0.0817
	28	0.1514	0.0790	0.0959

List of Figures

1 Design of output coding learning machines: monolithic (a) and parallel ensemble (b). 42

2 Performances of ECOC and OPC MLP, PLD and PND. (a) Monolithic MLP (b) PLD (c) PND. 43

3 Compared mutual information specific error matrix indices Φ_R between ECOC *monolithic* and *PND* learning machines on d5 (a), glass (b), optdigits (c) and letter (d) data sets. d5 is a synthetic data set available at `http:/ftp.disi.unige.it/person/ValentiniG/Data/d5.t`

4 Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *optdigits* data set. (a) 32 bit ECOC codewords using MLP base learners with 4 hidden units (b) 32 bit ECOC using MLP with 10 hidden units (c) 64 bit ECOC using MLP with 3 hidden units (d) 64 bit ECOC using MLP with 10 hidden units (e) 32 bit ECOC using linear perceptrons base learners (f) 64 bit ECOC using linear perceptrons base learners 45

5 Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *image-segmentation* data set. (a) 32 bit ECOC codewords using MLP base learners with 3 hidden units (b) 32 bit ECOC codewords using MLP base learners with 10 hidden units (c) 32 bit ECOC codewords using MLP base learners with 25 hidden units (d) 32 bit ECOC codewords using linear perceptron base learners 46

6 Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *p20* data set. (a) 50 bit ECOC codewords using MLP base learners with 6 hidden units (b) 50 bit ECOC codewords using MLP base learners with 12 hidden units (c) 50 bit ECOC codewords using MLP base learners with 24 hidden units (d) 50 bit ECOC codewords using linear perceptron base learners 47

7 Relationships between ECOC minimum Hamming distance and ensemble error using different base learners. (a) optdigits data set (32 bit codewords) (b) optdigits data set (64 bit codewords) (c) image-segmentation data set (d) p20 data set 48

8 Relationships between ECOC minimum Hamming distance among codewords, ensemble error and average base learner error with the *optdigits* data set. (a) 32 bit ECOC codewords using MLP base learners with 4 hidden units (b) 32 bit ECOC using MLP with 10 hidden units (c) 64 bit ECOC using MLP with 3 hidden units (d) 64 bit ECOC using MLP with 10 hidden units (e) 32 bit ECOC using linear perceptrons base learners (f) 64 bit ECOC using linear perceptrons base learners 49

9 Relationships between ECOC minimum Hamming distance between codewords, ensemble error and average base learner error with the *image-segmentation* data set. (a) 32 bit ECOC codewords using MLP base learners with 3 hidden units (b) 32 bit ECOC codewords using MLP base learners with 10 hidden units (c) 32 bit ECOC codewords using MLP base learners with 25 hidden units (d) 32 bit ECOC codewords using linear perceptron base learners 50

10 Relationships between ECOC minimum Hamming distance between codewords, ensemble error and average base learner error with the *p20* data set. (a) 50 bit ECOC codewords using MLP base learners with 6 hidden units (b) 50 bit ECOC codewords using MLP base learners with 12 hidden units (c) 50 bit ECOC codewords using MLP base learners with 24 hidden units (d) 50 bit ECOC codewords using linear perceptron base learners 51

11 Relationships between Average base learner error and ensemble error. Each point represents a different minimum Hamming distance between codewords. (a) optdigits: 64 bit ECOC codewords using MLP base learners (b) optdigits: 64 bit ECOC using linear perceptron base learners (c) p20: 50 bit ECOC using MLP base learners (d) image-segmentation: 32 bit ECOC using MLP base learners 52

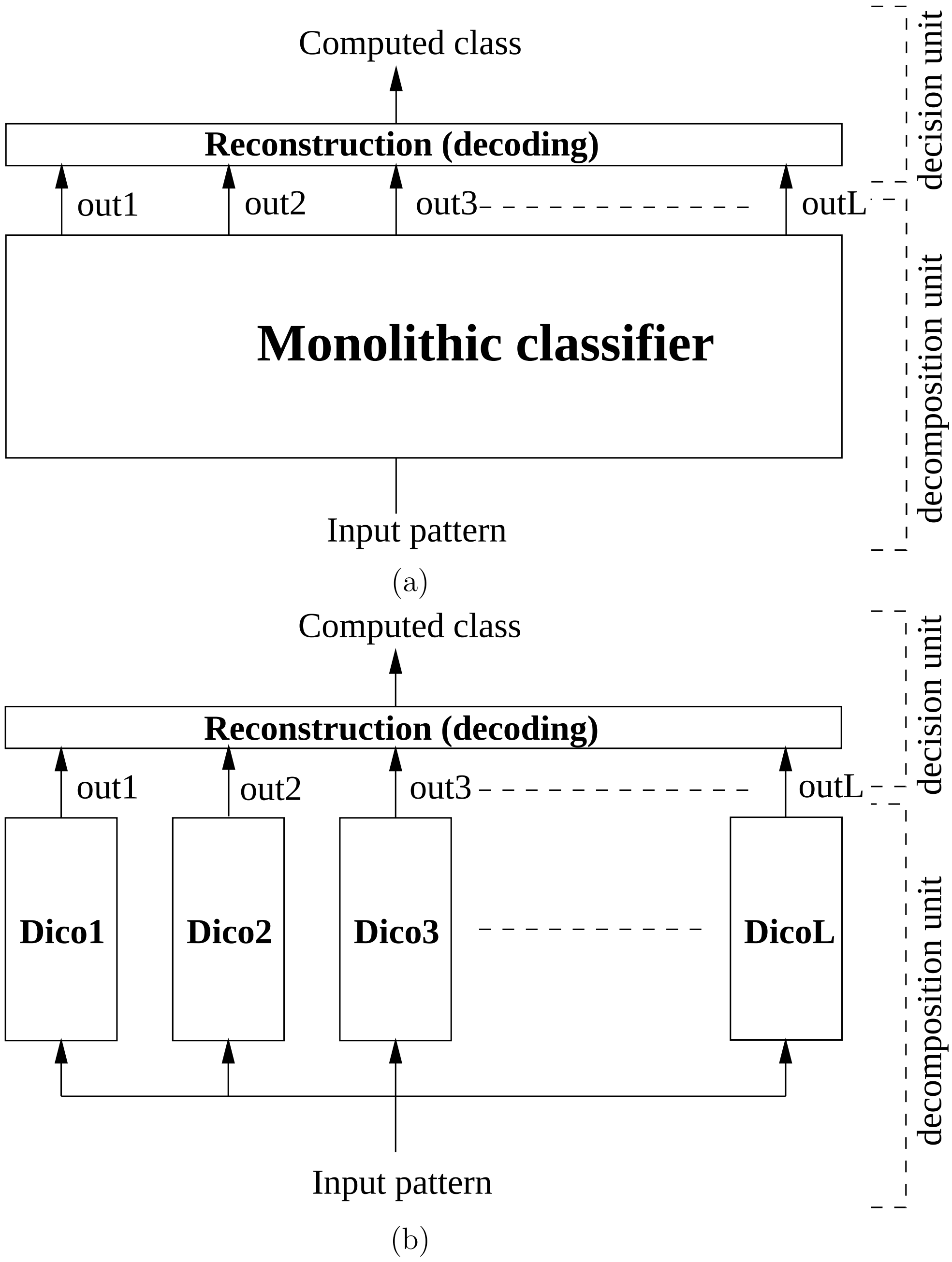
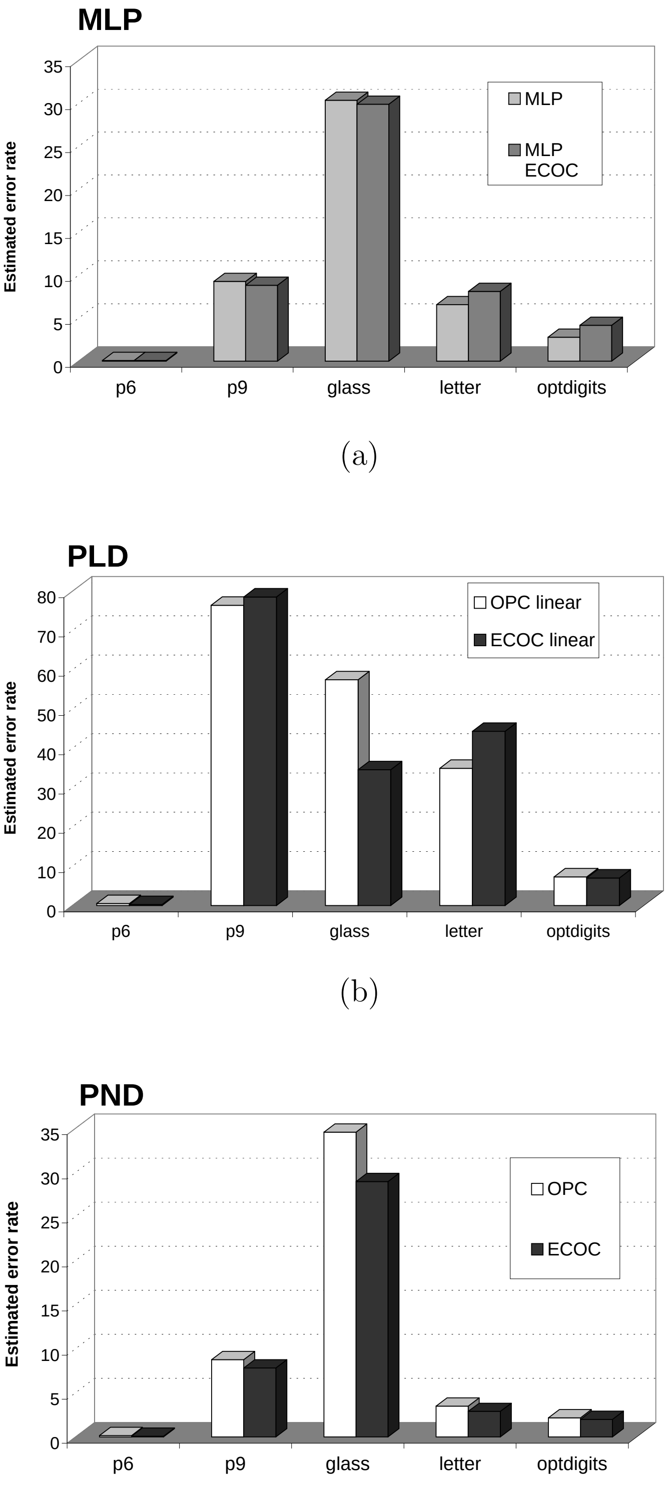


Figure 1: Design of output coding learning machines: monolithic (a) and parallel ensemble (b).

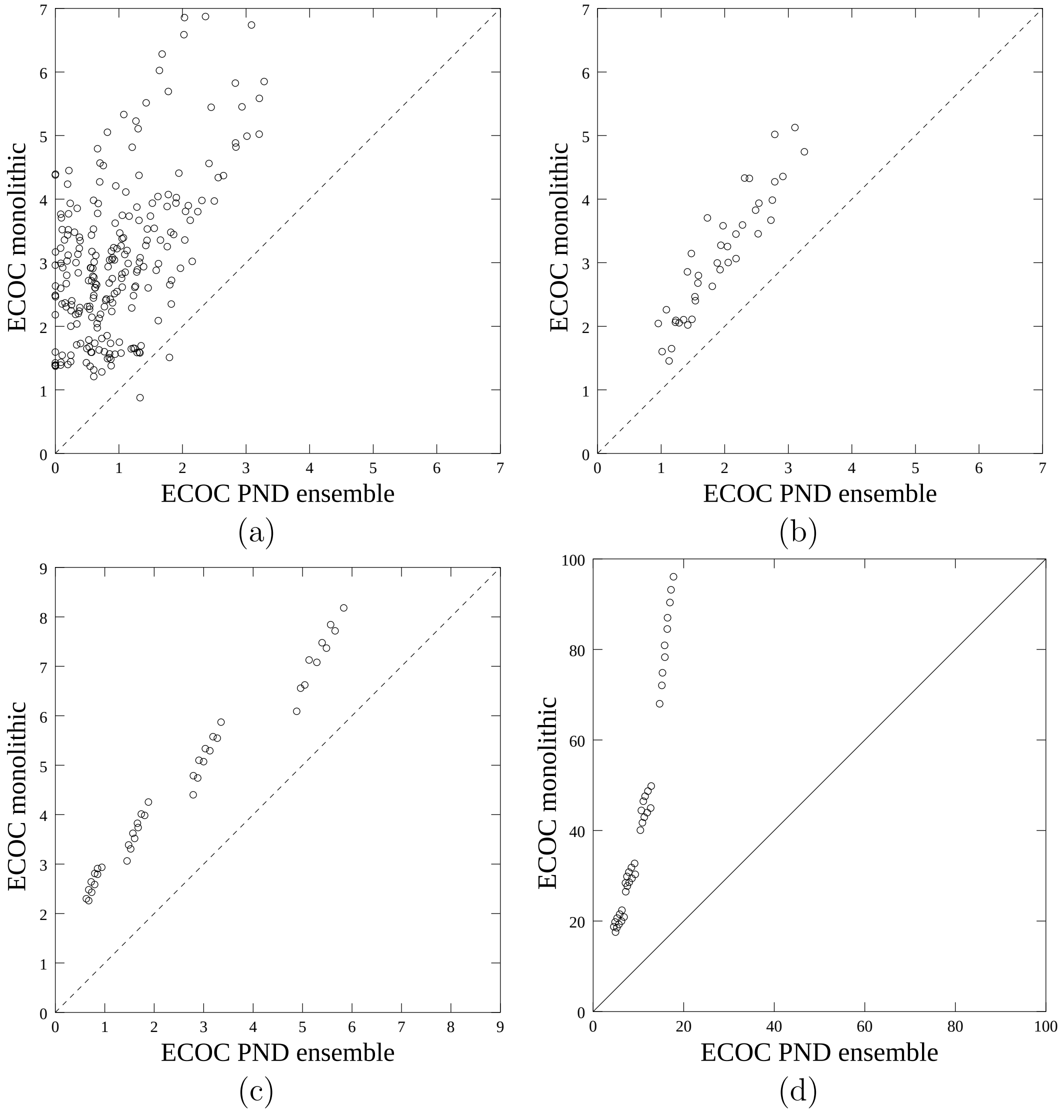


(a)

(b)

(c)

Figure 2: Performances of ECOC and OPC MLP, PLD and PND. (a) Monolithic MLP (b) PLD (c) PND.



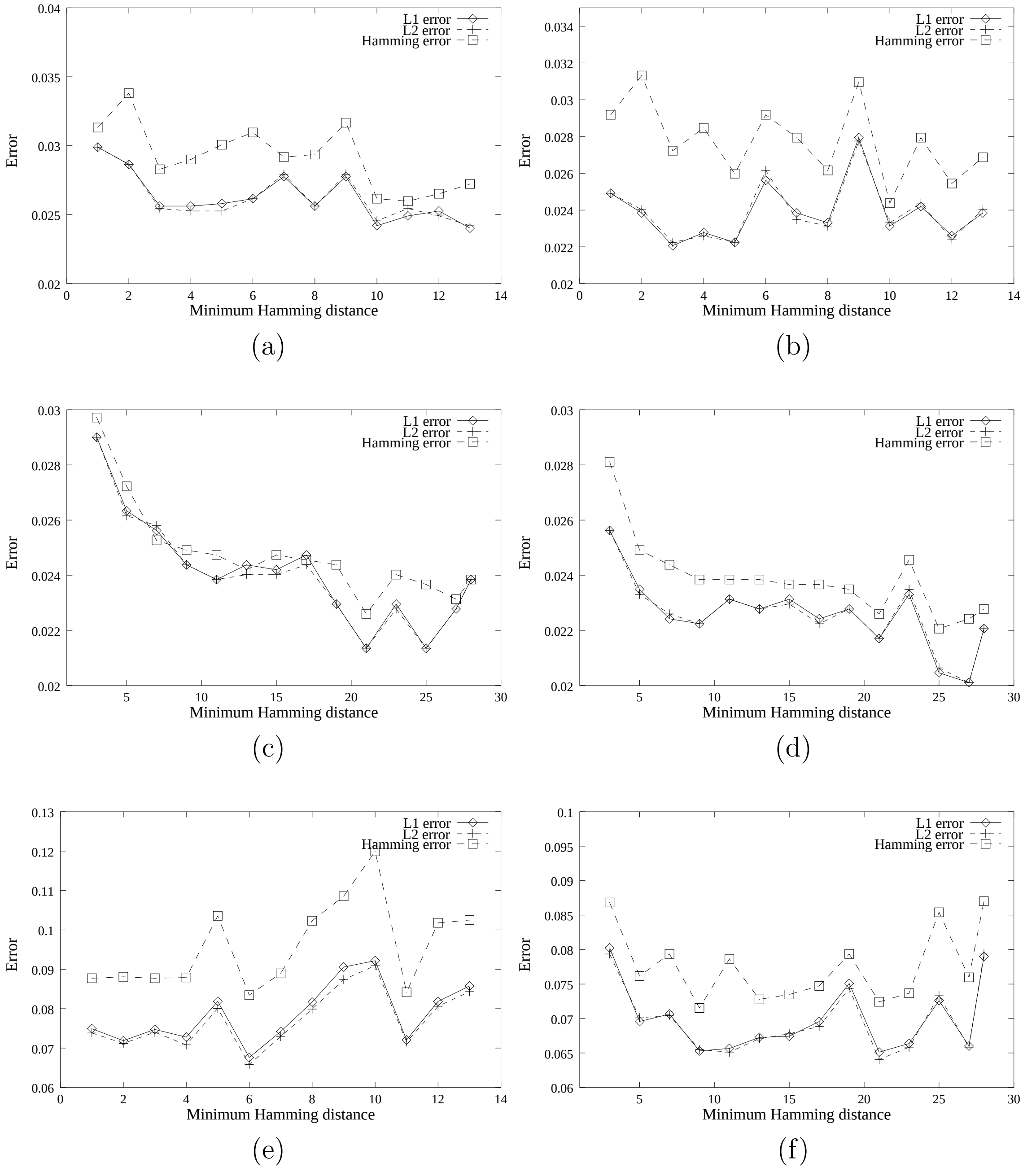
(a)

(b)

(c)

(d)

Figure 3: Compared mutual information specific error matrix indices Φ_R between ECOC *monolithic* and *PND* learning machines on d5 (a), glass (b), optdigits (c) and letter (d) data sets. d5 is a synthetic data set available at `http:/ftp.disi.unige.it/person/ValentiniG/Data/d5.tgz`

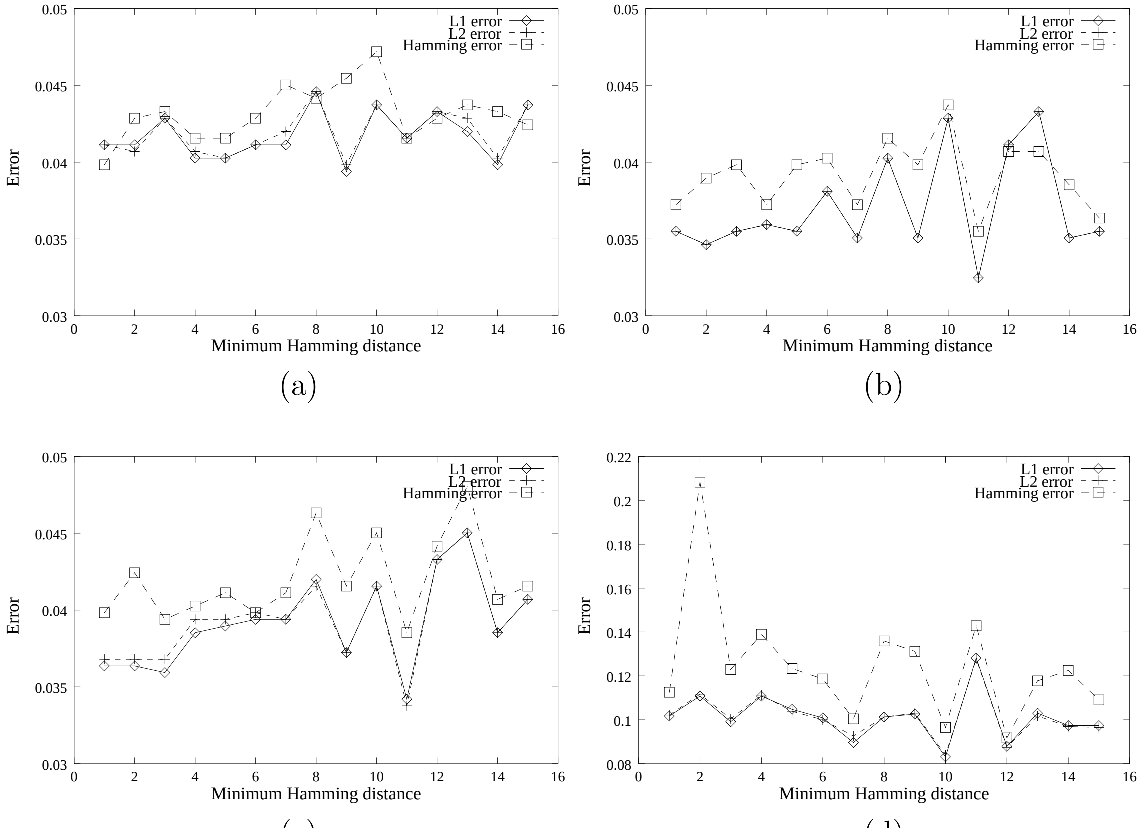


(a) (b)

(c) (d)

(e) (f)

Figure 4: Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *optdigits* data set. (a) 32 bit ECOC codewords using MLP base learners with 4 hidden units (b) 32 bit ECOC using MLP with 10 hidden units (c) 64 bit ECOC using MLP with 3 hidden units (d) 64 bit ECOC using MLP with 10 hidden units (e) 32 bit ECOC using linear perceptrons base learners (f) 64 bit ECOC using linear perceptrons base learners



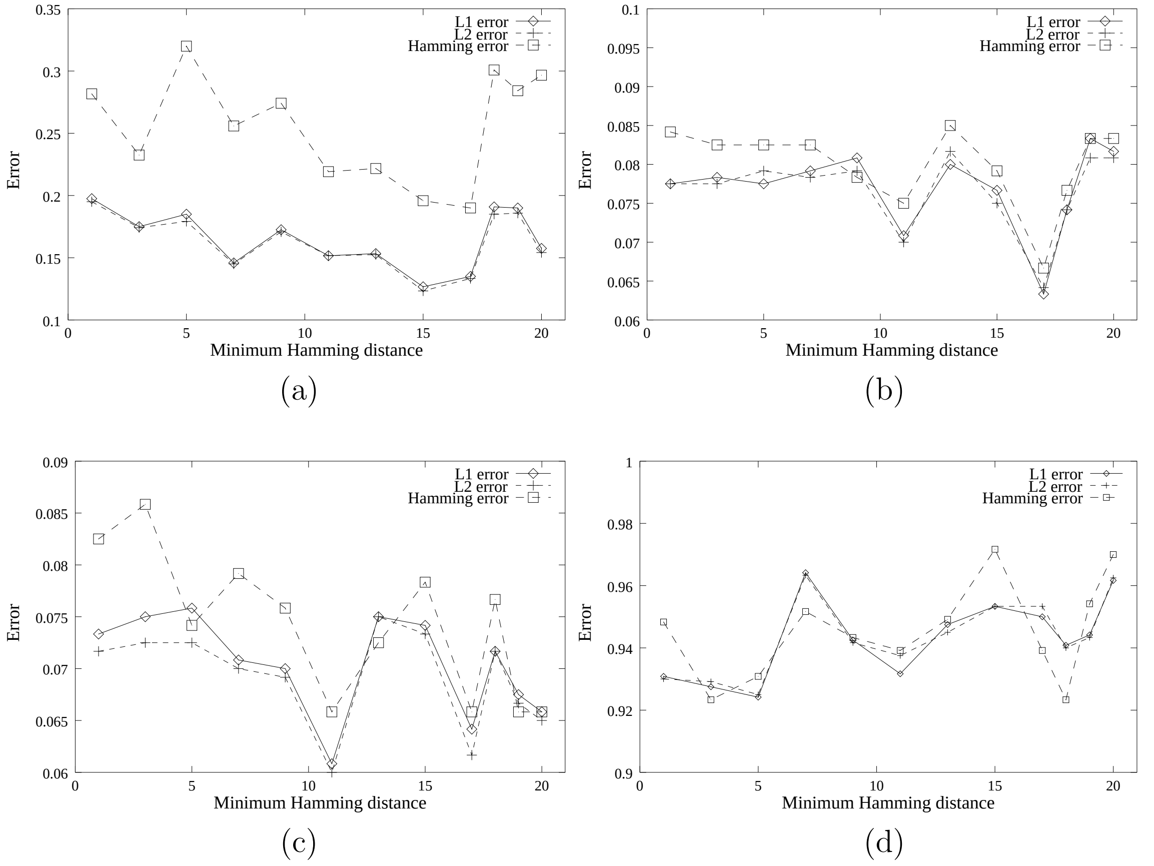
(a)

(b)

(c)

(d)

Figure 5: Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *image-segmentation* data set. (a) 32 bit ECOC codewords using MLP base learners with 3 hidden units (b) 32 bit ECOC codewords using MLP base learners with 10 hidden units (c) 32 bit ECOC codewords using MLP base learners with 25 hidden units (d) 32 bit ECOC codewords using linear perceptron base learners



(a) (b) (c) (d)

Figure 6: Relationships between ECOC minimum Hamming distance among codewords and ensemble errors using decoding functions based on L_1 and L_2 norm and Hamming distance with the *p20* data set. (a) 50 bit ECOC codewords using MLP base learners with 6 hidden units (b) 50 bit ECOC codewords using MLP base learners with 12 hidden units (c) 50 bit ECOC codewords using MLP base learners with 24 hidden units (d) 50 bit ECOC codewords using linear perceptron base learners

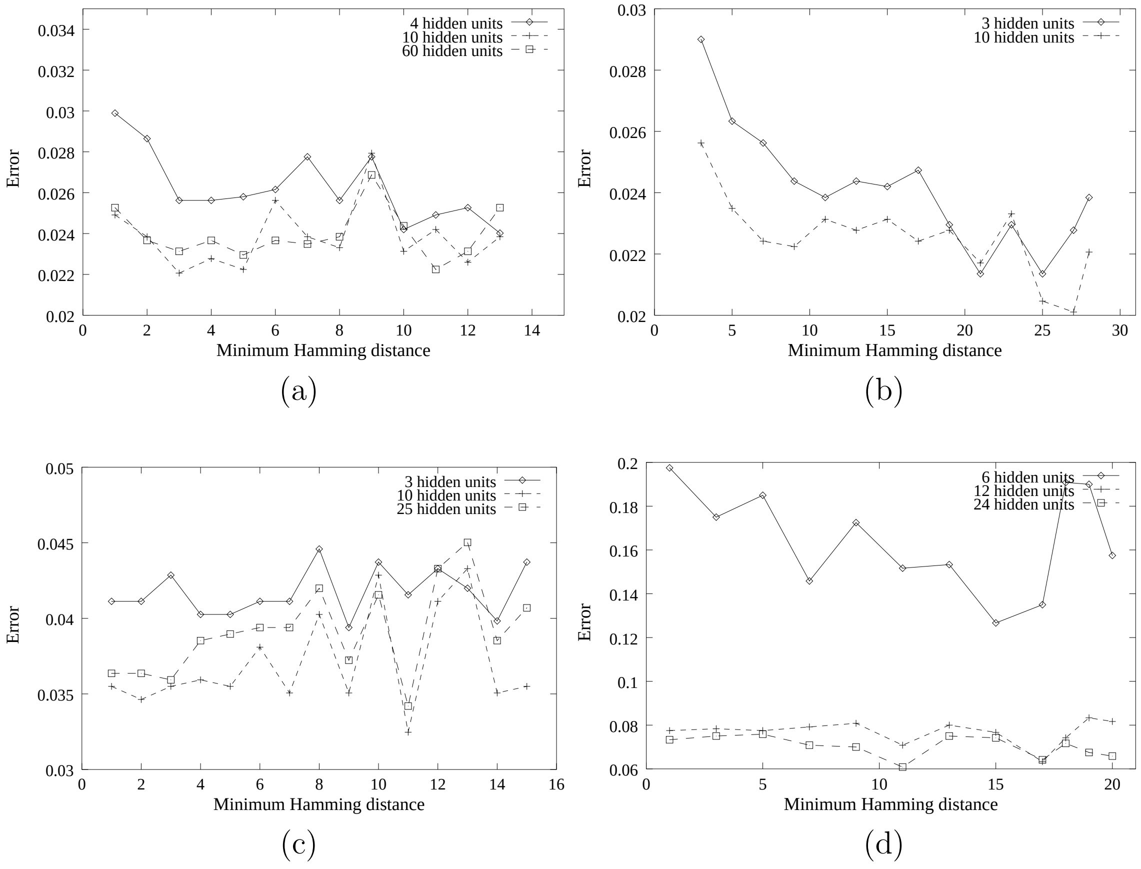
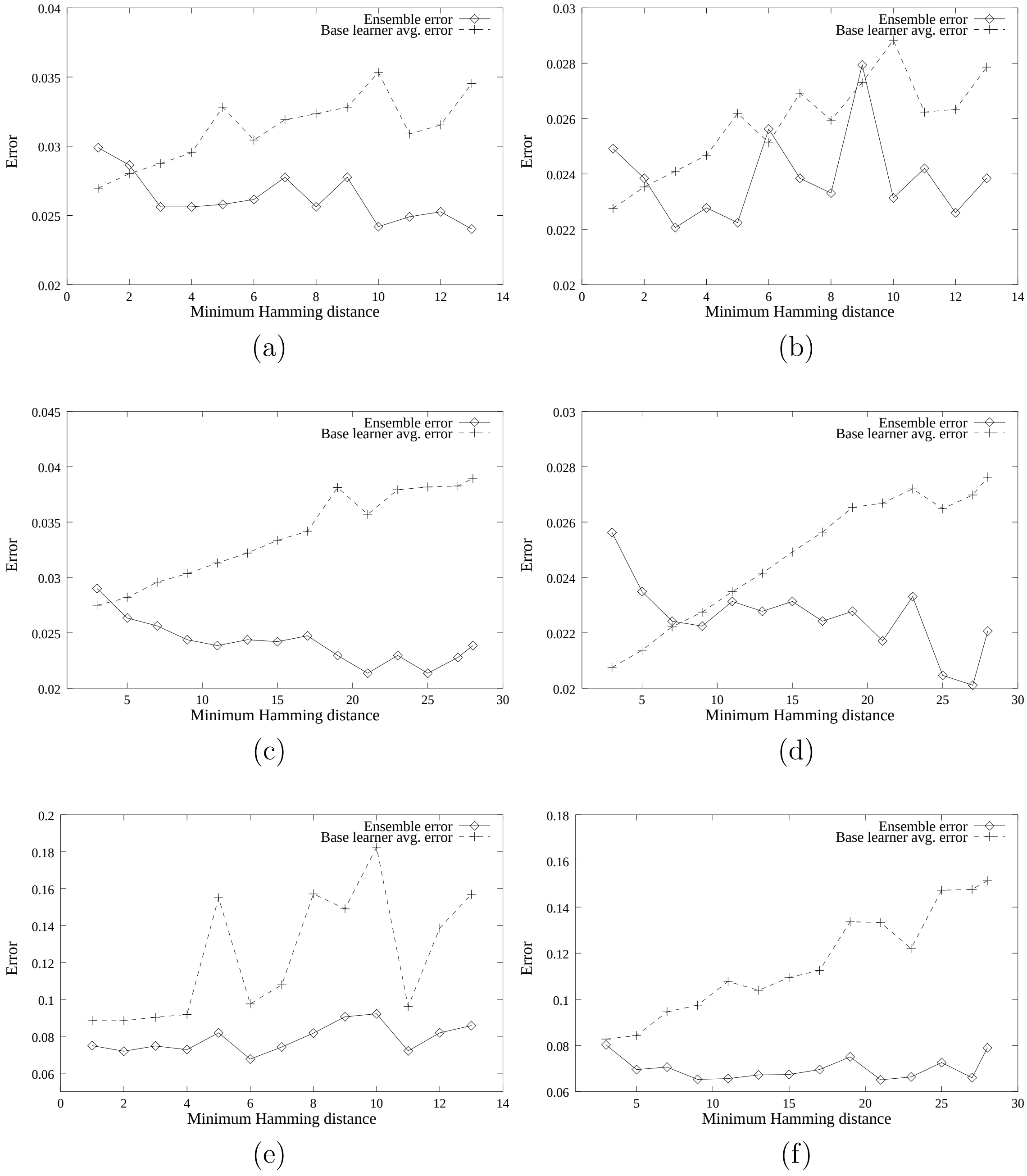


Figure 7: Relationships between ECOC minimum Hamming distance and ensemble error using different base learners. (a) optdigits data set (32 bit codewords) (b) optdigits data set (64 bit codewords) (c) image-segmentation data set (d) p20 data set



(a)

(b)

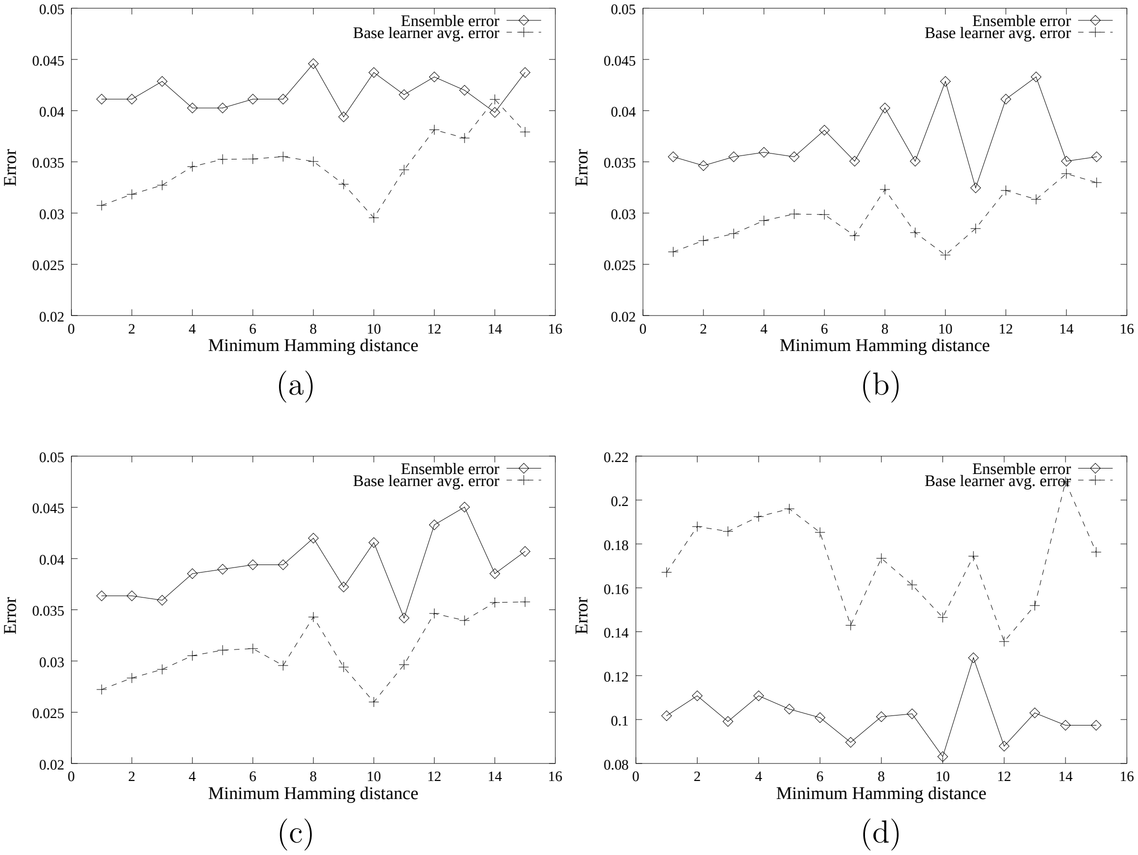
(c)

(d)

(e)

(f)

Figure 8: Relationships between ECOC minimum Hamming distance among codewords, ensemble error and average base learner error with the *optdigits* data set. (a) 32 bit ECOC codewords using MLP base learners with 4 hidden units (b) 32 bit ECOC using MLP with 10 hidden units (c) 64 bit ECOC using MLP with 3 hidden units (d) 64 bit ECOC using MLP with 10 hidden units (e) 32 bit ECOC using linear perceptrons base learners (f) 64 bit ECOC using linear perceptrons base learners



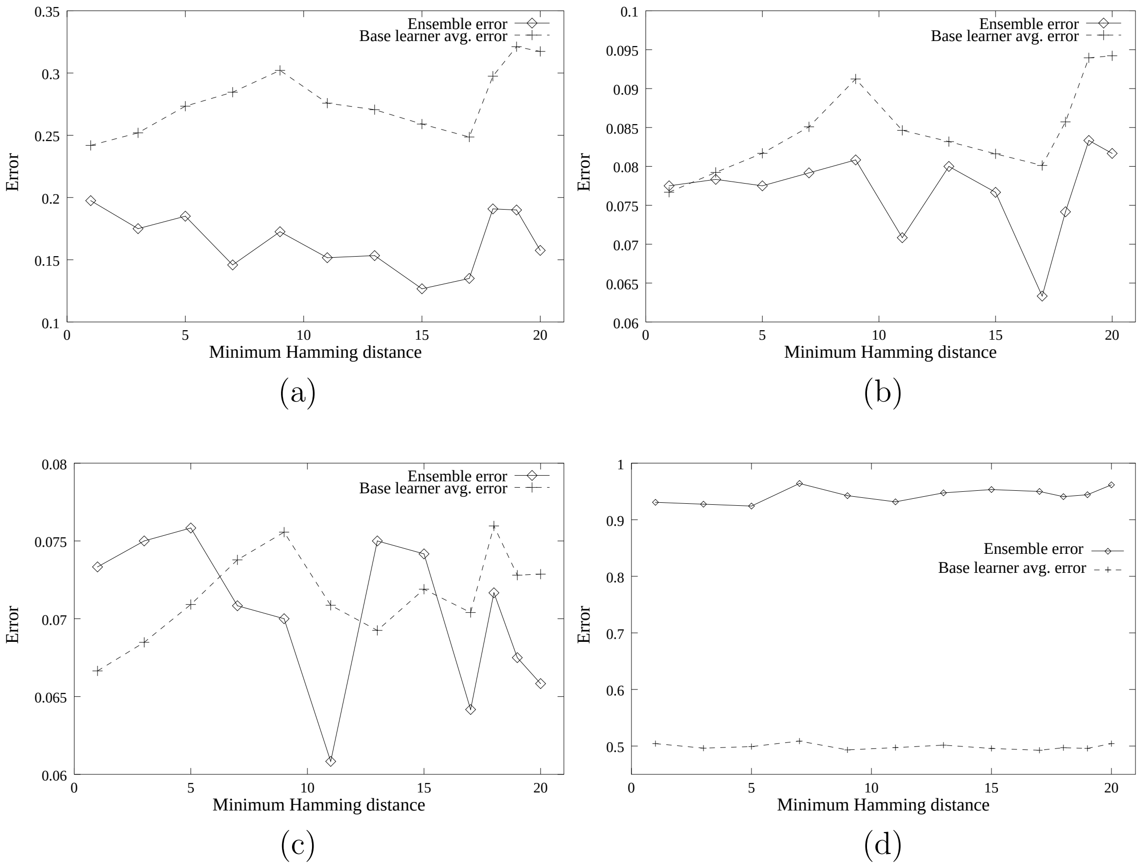
(a)

(b)

(c)

(d)

Figure 9: Relationships between ECOC minimum Hamming distance between codewords, ensemble error and average base learner error with the *image-segmentation* data set. (a) 32 bit ECOC codewords using MLP base learners with 3 hidden units (b) 32 bit ECOC codewords using MLP base learners with 10 hidden units (c) 32 bit ECOC codewords using MLP base learners with 25 hidden units (d) 32 bit ECOC codewords using linear perceptron base learners



(a)

(b)

(c)

(d)

Figure 10: Relationships between ECOC minimum Hamming distance between codewords, ensemble error and average base learner error with the *p20* data set. (a) 50 bit ECOC codewords using MLP base learners with 6 hidden units (b) 50 bit ECOC codewords using MLP base learners with 12 hidden units (c) 50 bit ECOC codewords using MLP base learners with 24 hidden units (d) 50 bit ECOC codewords using linear perceptron base learners

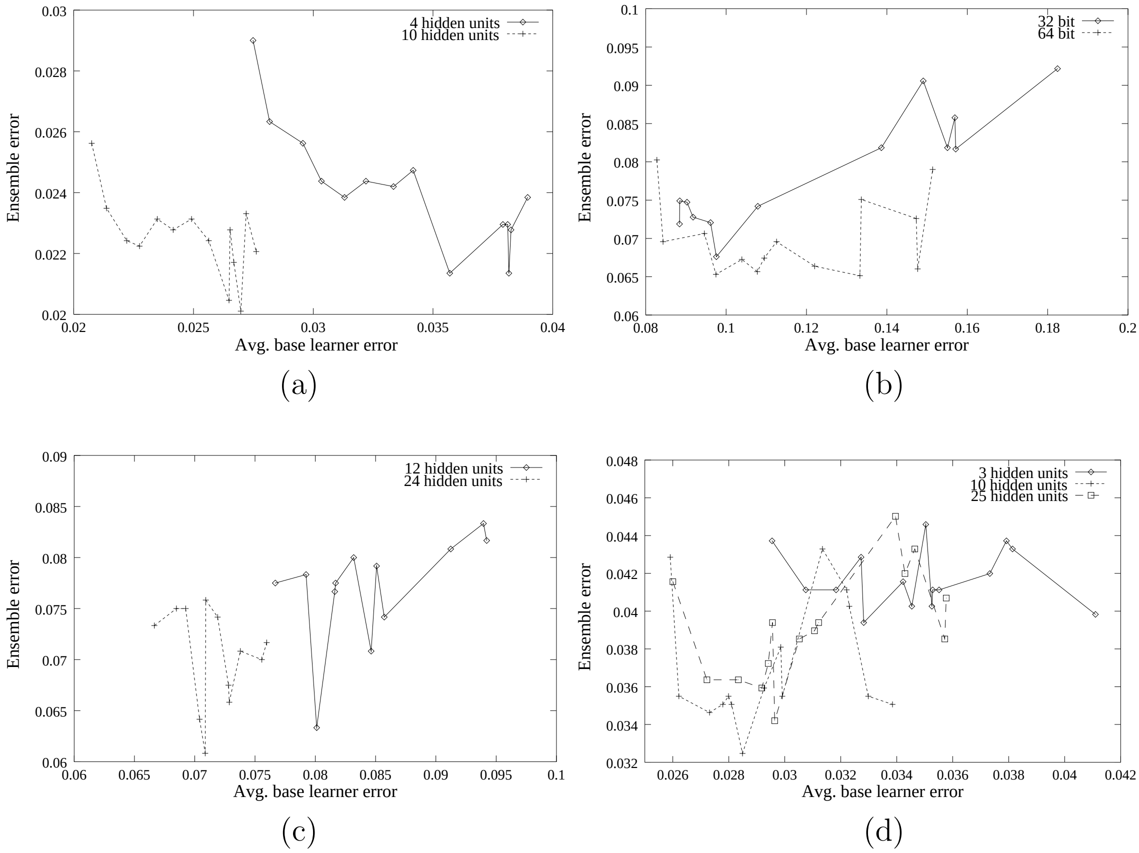


Figure 11: Relationships between Average base learner error and ensemble error. Each point represents a different minimum Hamming distance between codewords. (a) optdigits: 64 bit ECOC codewords using MLP base learners (b) optdigits: 64 bit ECOC using linear perceptron base learners (c) p20: 50 bit ECOC using MLP base learners (d) image-segmentation: 32 bit ECOC using MLP base learners