A Framework for Multiclass Reject in ECOC Classification Systems

Claudio Marrocco, Paolo Simeone, and Francesco Tortorella

DAEIMI, Università degli Studi di Cassino
Via G. Di Biasio 43, 03043 Cassino (FR), Italia
{c.marrocco,paolo.simeone,tortorella}@unicas.it

Abstract. ECOC is a diffused and successful technique to implement a multiclass classification system by decomposing the original problem in several two-class problems. In this paper we propose ECOC systems with a reject option carried out through two different schemes. The first one estimates the reliability of the output of the ECOC system and does not require any change in its structure. The second scheme, instead, estimates the reliability of the internal dichotomizers and implies a slight modification in the decoding stage. A final investigation is done on the sequential combination of both methods.

Keywords: ECOC, reject option, multiple classifiers systems.

1 Introduction

A diffused technique to face a classification problem with many possible classes is to decompose it into a set of two class problems. The rationale of this approach rely on the stronger theoretical roots and better comprehension characterizing two class classifiers (dichotomizers) such as Perceptrons or Support Vector Machines that, with this method, become employable in multiclass problems.

In this framework, Error Correcting Output Coding (ECOC) has emerged as a well established technique for many applications in the field of Pattern Recognition and Data Mining, mainly for its good generalization capabilities. In short, ECOC decomposition labels each class with a bit string (*codeword*) of length L, higher than the number of classes. The codewords are arranged as rows of a *coding matrix*, whose columns define each a two class problem; thus, for each problem, the set of the original classes parts into two complementary superclasses. On such problems induced by the coding matrix, L dichotomizers have to be trained in the learning phase. In the operating phase, the dichotomizers will provide a string of L outputs for each sample to be classified. The Hamming distance of such string from each of the codewords of the coding matrix is then evaluated and the class that corresponds to the nearest codeword is chosen. Usually, the codewords are chosen so as to have a high Hamming distance between each other; in this way, ECOC is robust to potential errors made by the dichotomizers. The reasons for the classification efficiency exhibited by ECOC seem to be the reduction of both bias and variance [1] and the achievement of

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 313–323, 2007.

a large margin [2]. After the seminal paper by Dietterich and Bakiri [3], many studies have been proposed which have analyzed several aspects of ECOC such as the factors affecting the effectiveness of ECOC classifiers [4], techniques for designing codes from data [5], evaluations of coding and decoding strategies [2].

A very common point in many applications in which the ECOC approach is used is that a classification error could have serious consequences, usually expressed by means of an error cost. In some cases, such cost can be so high that it is convenient to reject the sample (i.e. to suspend the decision and call for a further test) instead of risking a wrong decision. Obviously, also this choice involves a not negligible cost given by the charge of employing a more powerful system or requiring the decision of a human expert. Thus a rule is needed to find the optimal trade off between errors and rejects for the application at hand.

This paper proposes the introduction of a reject option for ECOC systems accomplished through two different schemes. The first one works on the output of the whole classification system and the reject is accomplished by considering the Hamming distance among the output codeword and the rows of the coding matrix. In the second scheme the reject option is performed on the base classifiers output by taking into account the confidence degrees provided by the dichotomizers. Such scheme makes use of a particular decoding technique for the erased bit in the codeword corresponding to rejects. To generalize the reject option, the cascade of the two approaches has been considered too.

In the rest of the paper we present, after a short description of the ECOC approach, the two schemes performing the reject option and the cascade of them. The successive section describes the results obtained from experiments performed on some UCI repository data sets. Some conclusions and future developments are drawn in the last section.

2 The ECOC Approach

The Error Correcting Output Coding has been introduced to decompose a multiclass problem into a set of complementary binary problems. Each class label is represented by a bit string of length L, called *codeword*, with the only requirement that distinct classes are represented by distinct codewords. If n is the number of the original classes, a code is a $n \times L$ matrix $\mathbf{C} = \{c_{hk}\}$ where $c_{hk} \in \{0, 1\}$. Each row of $\mathbf{C}$ corresponds to a codeword for a class, while each column corresponds to a binary problem. In this way, the multiclass problem is reduced to L binary problems on which L dichotomizers have to be trained. An example of coding matrix with $n = 5$ and $L = 12$ is shown in table 1. In the training phase, each dichotomizer is learned from a finite set of samples. In the operating phase, the sample $\mathbf{x}$ to be classified is fed to all the dichotomizers and each of them produces a binary value: all such values are collected to make a vector of binary decisions (*output vector*) to be compared with the codewords of the coding matrix. It is possible that some dichotomizer makes a wrong prediction, but this does not necessarily lead to an irrecoverable error in the multiclass problem since the code matrix is built by n distinct codewords of length $L > n$,

Table 1. An example of a coding matrix for a 5 classes problem

classes	codewords											
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
1	0	0	1	1	1	1	0	1	0	1	1	0
2	1	0	0	1	0	0	0	1	1	1	1	0
3	0	0	1	1	0	1	1	1	0	0	0	0
4	1	1	1	0	1	0	1	1	0	0	1	0
5	0	1	0	0	1	1	0	1	1	1	0	0

so as to make the Hamming distance between every pair of codewords as large as possible. The Hamming distance D_H between two words is given by the number of position where the bit patterns of the two words differ.

The minimum Hamming distance (MHD) $d = \min_{i,j} D_H(\mathbf{c}_i, \mathbf{c}_j)$ between any pair of codewords is a measure of the quality of the code. In particular it is possible to correct codewords which contains no more than $\lfloor (d-1)/2 \rfloor$ single bit errors. In this way, a single bit error does not influence the result, as it can happen when using the usual one-per-class coding, where the Hamming distance between each pair of strings is 2. To pass from the binary to the multiclass problem, the most common approach consists in evaluating the Hamming distances between the output vector $\mathbf{o}$ and the codewords of the matrix and choose for the nearest codeword, i.e. for the codeword exhibiting the minimum Hamming distance from the output vector. Therefore, the decision for the k-th class ω_k corresponding to the $\mathbf{c}_k$ codeword is taken according to:

$$\omega_k = \arg\min_j (D_H(\mathbf{c}_j, \mathbf{o})), \tag{1}$$

In particular, if the dichotomizers have soft output (e.g. they provide a confidence degree which is a real value ranging from 0 to 1), it is necessary to threshold their responses to obtain the value of the bits in the corresponding positions of the codeword.

3 The Multiclass Reject Option

The goal of this paper is to introduce a reject option for a multiclass problem in order to decrease the total classification cost by turning as many errors as possible into rejects. In fact, for a realistic problem, the error cost should be higher than a reject cost and thus an effective reject option is advantageous for the original multiclass classification problem. In general, a reject option is accomplished on a classifier by evaluating in some way the reliability of the decision taken by the classifier and rejecting the decision if the reliability is lower than a given threshold. In the case of an ECOC-based classification system, there are actually two places in which a decision is taken: the first place is the decoding stage, where the final multiclass decision is taken on the basis of the MHD. The second place is given by all the dichotomizers, each taking a two-class decision.

As a consequence, two different strategies are possible. The first one affects the decoding stage and evaluates the reliability of the multiclass decision on the basis of the MHD obtained; we will define *external* such scheme since it works at the output of the whole classification system. The second scheme (*internal* scheme), instead, evaluates the reliability of the outputs coming from the dichotomizers and rejects the decisions not sufficiently reliable. This approach affects the structure of the output vector since, in this case, it will contain, besides the usual values of 0 and 1, another symbol (let us call it r) which indicates that for the corresponding dichotomizer a reject has been taken. The decoding algorithm has to be consequently modified in order to handle the 3-value output vector. Obviously, this makes the second approach quite less general since, besides the change on the decoding stage, it puts some requirements on the characteristics of the dichotomizers to be employed. The two schemes are described in the following sections together with the cascade of them.

3.1 The External Reject Option

In literature, the decision in the ECOC approach has been principally based on the minimization of the Hamming distance among the codewords of the coding matrix and the output vector produced by the dichotomizers. Every employed dichotomizer gives an output that can be thresholded and combined to determine the final output vector: $\mathbf{o} = (o_1, o_2, \ldots, o_L)$.

Let us consider two codewords $\mathbf{c}_h$ and $\mathbf{c}_k$ that differs on d bits. If the number of erroneous bits is lower than $d/2$ we can correctly decode the word by using the MHD rule. When the number of errors is higher than $d/2$ it is not possible to recover the right codeword, i.e., the final decoding will be erroneous. This means that the greater is the Hamming distance between the output vector and the correct codeword the greater is the probability of an erroneous decision. In this situation it is possible to consider a reject rule based on the Hamming distance that introduces a reject region between the two codewords. This allows us to avoid to take a decision when the distance between the output vector and its nearest codeword is too high. If t_e is the reject threshold and ω_k is the class chosen according to eq. (1) the reject rule is:

$$r(\mathbf{o}, t_e) = \begin{cases} \omega_k & \text{if } D_H(\mathbf{c}_k, \mathbf{o}) < t_e, \\ reject & \text{if } D_H(\mathbf{c}_k, \mathbf{o}) \geq t_e. \end{cases} \tag{2}$$

Fig. 1 shows an example for such a problem. In fig. 1.a two samples belonging to the class ω_p produce two output vectors $\mathbf{o}_1$ and $\mathbf{o}_2$. In the first case a correct decision is taken while $\mathbf{o}_2$ will be assigned to the wrong class ω_q. Introducing the reject rule, a decision for the vector $\mathbf{o}_2$ will not be taken so avoiding an error (see fig. 1.b). It is worth noting that the lowest Hamming distance is zero while the highest one depends on the codewords of the matrix $\mathbf{C}$. If L is the maximum distance that we can have between two codewords (i.e., in the coding matrix there are two complementary rows) an upper bound of the maximum distance allowable for the reject threshold is $L/2$. It is worth noting that such scheme

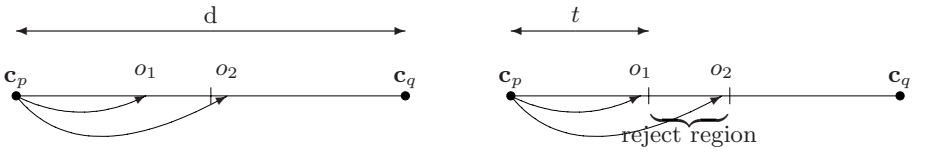

Fig. 1. Example of the decoding method based on the MHD in the standard approach (a) and with an external reject option (b)

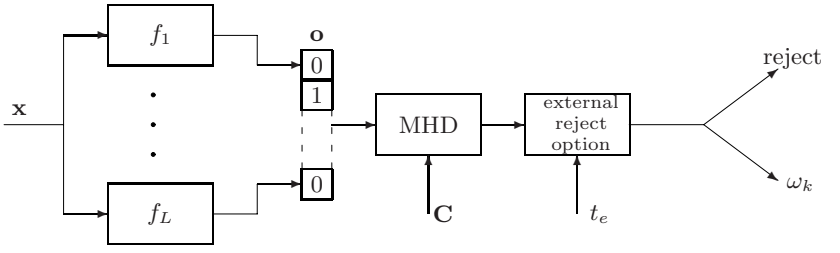

Fig. 2. The block diagram for the external reject rule

does not require any assumption neither on the dichotomizers nor on the coding matrix. The whole scheme is described in fig. 2.

3.2 The Internal Reject Option

Let us now suppose that we can estimate the reliability of the output of each dichotomizer in the ECOC system. For example, let us consider a model for the dichotomizer which provides a soft value ranging from 0 to 1. In this case, we should threshold the soft output to have a crisp response with a typical threshold value of 0.5. However, it is easy to see that a value for the soft output falling near the threshold will be much less reliable than a value near 0 or near 1. As a consequence, we can adopt a reject rule for each dichotomizers as:

$$o_j(\mathbf{x}, t_i) = \begin{cases} 1 & \text{if } f_j(\mathbf{x}) > 0.5 + t_i \\ 0 & \text{if } f_j(\mathbf{x}) < 0.5 - t_i \\ r & \text{otherwise} \end{cases} \tag{3}$$

Since in this case the output vector can also contain rejected bits, i.e. $c_i \in \{0, 1, r\}$, we have to focus on a decoding rule able to handle the 3 values. To this aim, it is possible to analyze the effect of an erasure (i.e. a reject) on the ECOC system. If μ is the number of erasures, the minimum distance between codewords (evaluated on the unerased bits) becomes $d-\mu$ and the error correcting capability of the code decreases to $\lfloor (d - \mu - 1)/2 \rfloor$. Therefore, to have a correct decision the number of errors and erasures should verify the following condition:

$$2\nu + \mu < d \tag{4}$$

where ν is the number of errors. This means that is twice difficult to correct an error than to correct an erasure. To show how the internal reject can be advantageous for the final decision, let us consider an output vector affected by ν errors. Without internal reject a correct decision will be taken if $2\nu < d$. Applying the internal reject rule we turn some erroneous bits (say μ_1) into erasures while the remaining erasures (say $\mu_2 = \mu - \mu_1$) come from correct decisions. In this case, the correct decision will be taken if $2\nu_1 + \mu_1 + \mu_2 < d$ where $\nu_1 = \nu - \mu_1$. Therefore, we will take advantage from the internal reject if $\mu_1 < \mu_2$ that is if at least half of the erasures comes from erroneous bits.

In order to take a decision, an erasure filling method called *erasure decoding* [6] is adopted in the decoding stage. To understand its rationale, let us suppose to replace all the erased bits by 0 and decode the obtained vector. If no more than half of the erasures should have been ones and eq. (4) is satisfied, then the number of errors is still less than half of d and the decoding will be correct. On the other hand, if more than half of the erasures should have been ones then we are introducing other bit errors and the decision will be erroneous. In this case, if we fill all the erased bits with 1 the decision will be successful. Therefore, the erasure filling procedure consists in decoding twice and choose the codeword that is closer to the output vector in terms of Hamming distance. The resulting procedure can be summarized as follows:

1. Place zeros in all erased position and decode to the closer codeword (in Hamming distance terms) $\mathbf{c}^{(0)}$;
2. Place ones in all erased position and decode to the closer codeword (in Hamming distance terms) $\mathbf{c}^{(1)}$;
3. Choose the closest $\mathbf{c}^{(j)}$ to the received codeword in the unerased positions, where $j = 0, 1$.

The first two steps of the algorithm are meant to solve the rejects/erasures while the last one exploits the error correction capability of the code.

However, it could happen that the output vector falls (according to the erasure decoding) on the halfway between two different codewords. In this case, the decision can not be reliably taken and thus a reject is produced. The complete rule can be described as:

$$r(\mathbf{C}, \mathbf{x}) = \begin{cases} \omega_k^{(0)} & \text{if } D_H^*(\mathbf{c}_k, \mathbf{c}^{(0)}) < D_H^*(\mathbf{c}_k, \mathbf{c}^{(1)}), \\ \omega_k^{(1)} & \text{if } D_H^*(\mathbf{c}_k, \mathbf{c}^{(1)}) < D_H^*(\mathbf{c}_k, \mathbf{c}^{(0)}), \\ reject & \text{if } D_H^*(\mathbf{c}_k, \mathbf{c}^{(0)}) = D_H^*(\mathbf{c}_k, \mathbf{c}^{(1)}). \end{cases} \tag{5}$$

where D_H^* is the Hamming distance on the unerased bits. The resulting system is shown in fig. 3.

3.3 The Cascade Reject Option

It is worth noting that the output of the ECOC system provided with the internal reject option is still based on the MHD criterion. Therefore, it is possible to implement a cascade of the two procedures before described using the output

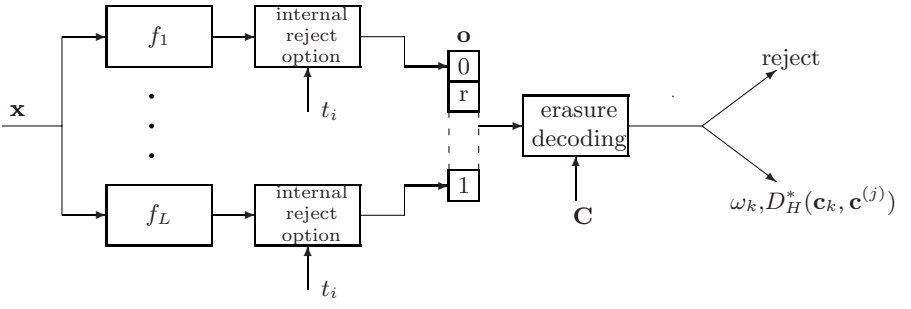

Fig. 3. The block diagram for the internal reject rule

of the internal rule as input for the external reject option. In such a case the Hamming distance between the codewords and the output vector (and then the threshold for the external reject option) is evaluated only on the unerased bits. The goal of the cascade of the two methods is to reduce the number of erroneous decision that we obtain after the internal rule. It is worth noting that in this case we have to choose two different thresholds. A block scheme of this approach (that we called *cascade reject rule*) is reported in fig. 4.

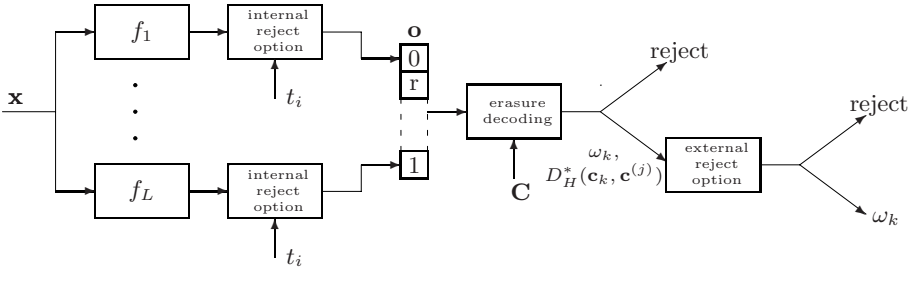

Fig. 4. The block diagram for the cascade reject rule

4 Experiments

In order to evaluate the performance of the proposed methods, experiments were made on some data sets publicly available at the UCI Machine Learning Repository [7]; all of them have numerical input features and a variable number of

Table 2. Data sets and coding matrices used in the experiments

Data Sets	# Classes	# Features	Coding Matrix	Length (L)	# Samples
Glass	6	9	Exhaustive	31	214
SatImage	6	36	Exhaustive	31	6435
Yeast	10	8	BCH 31-21	31	1484
Vowel	11	10	14-11	14	435

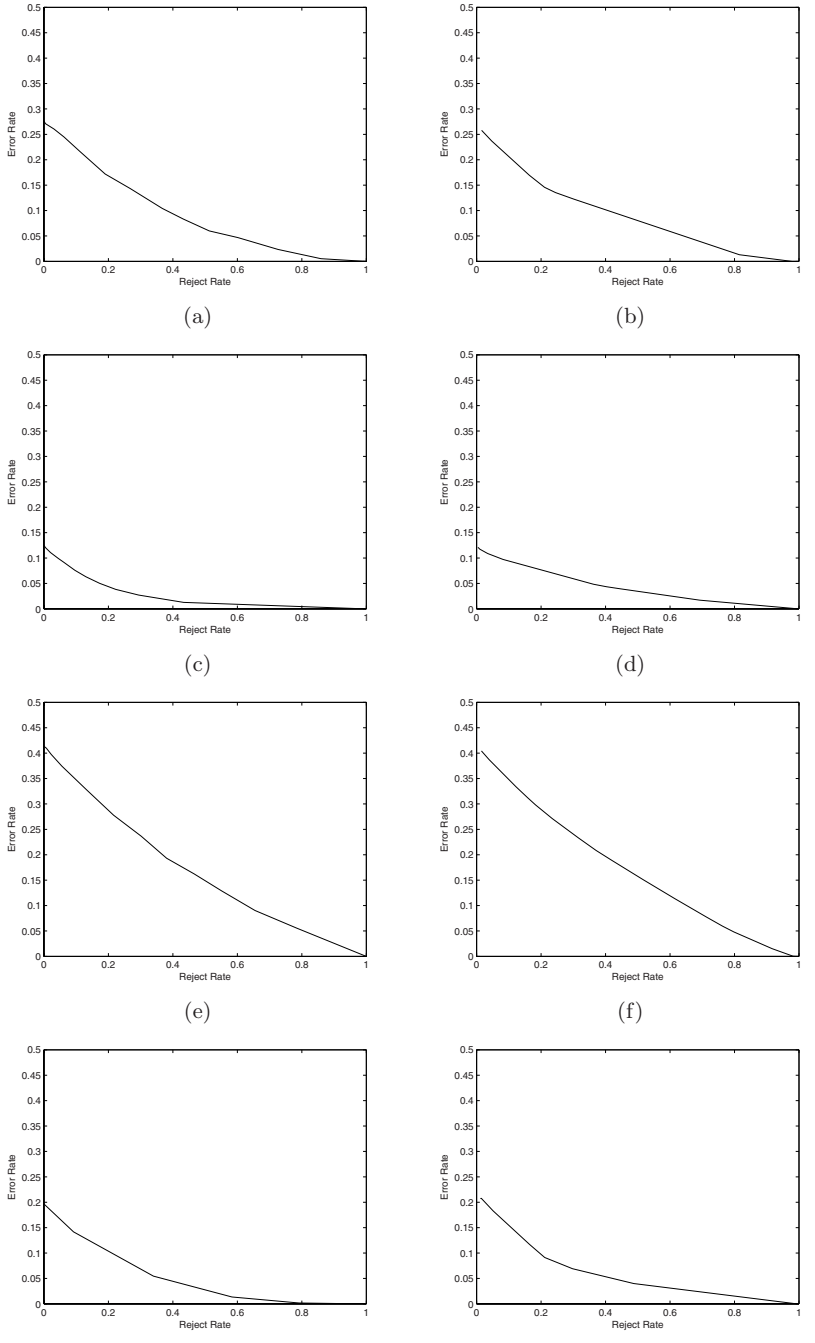

Fig. 5. Comparison between external (left side) and internal (right side) reject option on the different data sets: (a-b) Glass, (c-d) SatImage, (e-f) Yeast, (g-h) Vowel

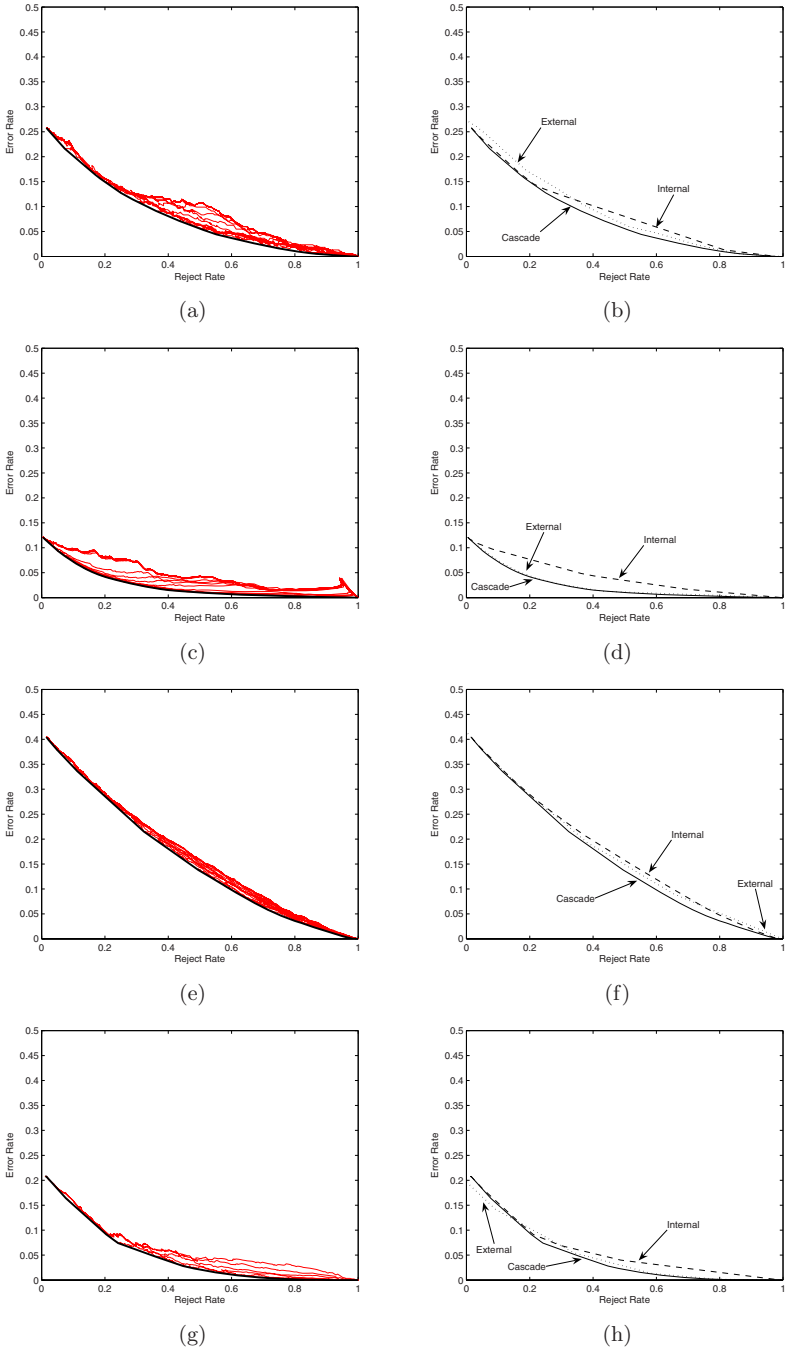

Fig. 6. The results obtained with the cascade option (left side) and the comparison between the three methods (right side) on the different data sets: (a-b) Glass, (c-d) SatImage, (e-f) Yeast, (g-h) Vowel

classes. More details for the data sets are given in table 2. The table provides also the type of ECOC matrix employed for each data set. As in [3] we have chosen an exhaustive code for the sets that have a number of classes lower than 8 and a BCH code for those having a number of classes greater than 8. In particular, for Vowel data set we used a matrix (named 14-11) with a reduced number of columns available at `http://web.engr.oregonstate.edu/~tgd/software/ecoc-codes.tar.gz` As base dichotomizers in the ECOC framework Modest AdaBoost [8] has been used using a simple decision tree as weak learner with a randomized number of splits in every run. To avoid any bias in the comparison, 12 runs of a multiple hold out procedure have been performed on all the data sets. In each run, the data set has been split in three subsets: a training set (containing the 70% of the samples of each class) to train the base classifiers, a validation set and a test set (each containing the 15% of the samples of each class) used respectively to normalize the outputs into the range $[0, 1]$ and to evaluate the performance for the multiclass classification.

To compare the different methods a useful representation to evaluate the benefits of a reject option is the error-reject curve that has been built varying the opportune thresholds t_i and t_e for all the data sets. In fig. 5 we report the results of the comparison between the external and internal schemes. The number of reject thresholds for the two cases are different: the external approach considers values ranging between $[0, L/2]$ as discussed in section 3.1 while the internal rule considers all the possible normalized output values observed in the range $[0, 0.5]$. It should be also noted that since for the internal option we fix a multiclass reject rule (see eq. 5) we obtain a reject rate always greater than zero since we can have a reject even if $t_i = 0$. Experimental results does not show better performance of one of these strategies on the other but they are practically equivalent. In fig. 6 we show (on the left side) the results obtained on each data set with the cascade approach. In each graph the error-reject curves varying the internal threshold for a fixed external threshold are reported. For the sake of comparison the convex hull of all these curves has been evaluated and compared with the two previous methods in the right side of fig. 6. The cascade option presents always a lower error-reject curve on all the data sets with only one exception on Vowel data set (see fig. 6) where for the range $[0, 0.18]$ the curve of the external reject rule exhibits lower error probabilities.

5 Conclusions

In this paper we have proposed two schemes to provide an ECOC classification system with a reject option. The experiments have shown that the two methods give similar results, even though they are effective on different situations. In fact, when both are activated in a cascade scheme, the results obtained are clearly better. The future work will focus on the analysis of particular codes more suitable for erasure decoding.

References

1. Kong, E.B., Dietterich, T.G.: Error-Correcting Output Coding Corrects Bias and Variance In: International Conference on Machine Learning, pp.313–321 (1995)
2. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 9–16 (2000)
3. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
4. Masulli, F., Valentini, G.: An experimental analysis of the dependence among codeword bit errors in ECOC learning machines. Neurocomputing 57, 189–214 (2004)
5. Pujol, O., Radeva, P., Vitrià, J.: Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. IEEE Trans. On Pattern Analysis And Machine Intelligence 28(6), 1001–1007 (2006)
6. Morelos-Zaragoza, R.H.: The Art of Error Correcting Coding. Wiley & Sons, Chichester (2002)
7. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases, `www.ics.uci.edu/~mlearn/MLRepository.html` (1998)
8. Vezhnevets, A., Vezhnevets,V.: Modest AdaBoost - Teaching AdaBoost to Generalize Better. Graphicon-2005, Novosibirsk Akademgorodok, Russia, `http://graphics.cs.msu.su/en/research/boosting/index.html` (2005)