On ECOC as Binary Ensemble Classifiers

J. Ko[1] and E. Kim[2]

[1] Dept. of Computer Engineering, Kumoh National Institute of Technology
1, Yangho-dong, Gumi, Gyeongbuk 730-701, Korea
nonezero@kumloh.ac.kr,
[2] National Computerization Agency,
NCA Bldg 77, Mugyo-dong, Jung-gu, Seoul 100-775, Korea
outframe@nca.go.kr

Abstract. The Error-Correcting Output Codes (ECOC) is a representative approach of the binary ensemble classifiers for solving multi-class problems. There have been so many researches on an output coding method built on an ECOC foundation. In this paper, we revisit representative conventional ECOC methods in an overlapped learning viewpoint. For this purpose, we propose new OPC based output coding methods in the ECOC point of view, and define a new measure to describe their properties. From the experiment on a face recognition domain, we investigate whether a problem complexity is more important than the overlapped learning or an error correction concept.

1 Introduction

The Error-Correcting Output Codes (ECOC) [1] is one of the binary ensemble classifiers for solving multi-class problems. The ECOC has been dominant theoretical foundation in output coding methods [2-6] that decompose a complex multi-class problem into a set of binary problems and then reconstructs the outputs of binary classifiers for each binary problem. The performance of output coding methods depends on base binary classifiers. It needs to revisit the ECOC concept, since the Support Vector Machines (SVM) [7] that can produce a complex nonlinear decision boundary with a good generalization performance is available as a base classifier for output coding methods.

The ECOC has two principals with respect to a codes design in which the codes concern both how to decompose a multi-class problem into several binary ones and how to decide a final decision. One principal is to enlarge the minimum hamming distance of a decomposition matrix. The other is to enlarge the row separability to increase the diversity among binary problems. A high diversity reduces an error-correlation among binary machines [8]. By enlarging the length of codewords [9], we can easily increase the hamming distance of the decomposition matrix at the cost of generating a large number of binary problems. In this circumstance, each class can be learned redundantly in several binary machines, we call it *overlapped learning*. By increasing the error-correction ability through the overlapped learning, we have been able to improve performance of a conventional ECOC with a hamming decoding. The

hamming decoding closely concerns the hamming distance of the decomposition matrix.

In a *generalized ECOC* [9] that includes 0 elements as well as –1 and +1 in the decomposition matrix, i.e., it has a triple codes (on the other side, a *conventional ECOC* consists of -1 and +1, i.e., it has a binary codes), we cannot directly compute the hamming distance. A new distance, a generalized hamming distance, is defined by [9], where the distance between the 0 element and the others is 0.5. The primary motivation of the conventional ECOC has been the overlapped learning of classes built on binary codes. The generalized ECOC does not insist on the binary codes any more, and the SVM used for a binary classifier can produce a real-valued confidence output that can be useful information for discriminating classes.

In this paper, we revisit ECOC with respect to the generalized ECOC by comparing and empirically analyzing certain properties of the representative ECOC methods, such as One-Per-Class (OPC) [11], All-Pairs [12], Correcting Classifier (CC) [10] and our proposed OPC-based methods designed on conventional ECOC concept. Further, we give an empirical conclusion on a codes design, which is limited to our experiment on face recognition.

2 One-Per-Class Variants with ECOC concept

In this section, we firstly formulate the *output coding* method (a generalized ECOC) in two steps: decomposition and reconstruction. Then, we propose new OPC based output coding method with ECOC concept, and define a new measure to describe their properties. Further, we describe later the performance of ECOC *with margin decoding*, which uses the real-valued output of a machine, using a newly defined problem complexity measure in the experiment. The OPC with hamming decoding has no error correction ability, so we begin by introducing additional machines to endow it with an error correcting ability.

2.1 Decomposition and Decoding

Decomposition (Encoding): A decomposition matrix, $D \in \{-1,\ 0,\ +1\}^{L \times K}$, specifies K classes to train L machines (dichotomizers), $f_1,\ldots, f_L$. The machines f_l is trained according to the row $D(l,\cdot)$. If $D(l,k) = +1$, all examples of class k are positive and if $D(l,k) = -1$, all examples of class k are negative, and if $D(l,k) = 0$ none of the examples of class k participates in the training of f_l. The column of D is called *codewords*. The entry "0" is introduced by [9]. Hence, some examples for $D(l,k) = 0$ can be omitted in the training phase.

We can formulate two separated super classes C_l^+ and C_l^- for the machine f_l as follows: $C_l^+ = \{C_k \mid D(l,k) = 1\}$, $C_l^- = \{C_k \mid D(l,k) = -1\}$.

Decoding (Reconstruction): In the decoding step, a simple nearest-neighbor rule is commonly used. The class output is selected that maximizes some similarity measure $s : \mathbf{R}^L \times \{-1, 0, 1\}^L \rightarrow [-\infty, \infty]$, between $f(\mathbf{x})$ and column $D(\cdot, k)$.

$$class\_output = \arg\max_k s(f(\mathbf{x}), D(\cdot, k)) \quad (1)$$

We call it a *margin decoding*, equation (2), a similarity measure based on a *margin*, defined as $y \cdot f(x)$ [9]. When classifier outputs a hard decision, $h(\mathbf{x}) \in \{-1, 1\}$, the method is called *hamming decoding*, equation (3).

$$s(f(\mathbf{x}), D(\cdot, k)) = \sum_l f_l(\mathbf{x}) D(l, k) \quad (2)$$

$$s_H(h(\mathbf{x}), D(\cdot, k)) = 0.5 \times \sum_l (1 + h_l(\mathbf{x}) D(l, k)) \quad (3)$$

2.2 New Decompositions

Tree-Based Decomposition: We design the tree structure for getting additional machines as well as those of generated by OPC. We adopt binary tree and distribute the classes of a parent node to its child nodes in a crossing manner. By the crossing manner, we can achieve the diversity of the binary problems with our proposed decomposing method as follows. Each node except for the root node makes one row in a decomposition matrix by assigning a positive value for classes that the node has, and a negative value for the other classes in the sibling nodes. The root node gives a positive value for the half of the whole classes and a negative value for the remainder. Fig. 1 shows a generated decomposition tree and a decomposition matrix on 8 classes.

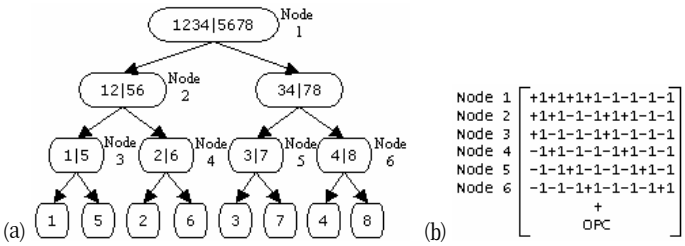

Fig.1. Decomposition matrix of Tree-based scheme for 8 classes. (a) tree-structure on *8* classes. (b) Its decomposition matrix.

When the number of classes is K, the $2 \times (K - 1)$ problems are generated. The difference between the number of classes being a positive class and the number of classes being a negative class varies according to the level of depth of the tree, so each binary problem can have the different level of complexity. Therefore, it is desirable to introduce weights into the decoding process to handle a different complexity among problems.

***N*-Shift Decomposition**: In this scheme, we first decide the number of positive classes N, and then form the first row of a decomposition matrix by setting N elements from left as positive ones and the remainder as negative ones. The rest rows are

easily constructed by right-shifting the elements of the preceding row. Finally, OPC decomposition matrix is added to it. When the number of classes is K, the $2 \times K$ problems are generated. . Fig. 2 shows two examples of a generated decomposition matrix having different N values, 2 and 3, respectively, when K is 4.

$$\begin{bmatrix} +1 & +1 & -1 & -1 \\ -1 & +1 & +1 & -1 \\ -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 \\ & + & & \\ & \text{OPC} & & \end{bmatrix} \qquad \begin{bmatrix} +1 & +1 & +1 & -1 \\ -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 \\ & + & & \\ & \text{OPC} & & \end{bmatrix}$$

(a) $N = 2$ (b) $N = 3$

Fig. 2. Decomposition matrix of *2*-Shift and *3*-Shift for 4 classes.

2.3 New Decodings

It is undesirable to deal with the outputs of the machines equally where each machine is trained with a problem having different level of complexity. There are two possible solutions to this problem: One is to utilize the different level of output for class decision, and the other is to adopt a weighed output. In this section, we propose the *relative distance decoding* for the former, and the *weighted decoding* for the latter respective.

Relative Distance Decoding: The machine has different scale outputs for two classes, so the same outputs should be understood differently. As an example, consider that, for samples belonging to class i, the machine habitually generates *0.8*, and for samples belonging to class j, *0.5*. The habit of generating uneven outputs for classes is formed during the learning process, and can be used for discriminating classes. To utilize this information, we introduce an *average template*. The average template is constructed by calculating the average of output for each machine as follows:

$$D'(i, j) = \left(\sum_{\mathbf{x} \in C_j} f_i(\mathbf{x})\right) / \left|C_j\right| \tag{4}$$

where $\left|C_j\right|$ means the number of samples belonging to the class j. The following equation calculates the similarity between a given input and a considered class by the relative distance.

$$rd(f(x), D'(\cdot, k)) = 1/(1 + \exp(Ad + B)) \tag{5}$$
$$d = \sum_l \left\| f_l(\mathbf{x}) - D'(l, k) \right\|_2$$

Both *A* and *B* constants of the exponential function and they can be usually fixed by experiment.

Weighted Decoding: As the number of positive classes increases, the complexity of the binary problem increases accordingly. There is a difference between the confidences on the outputs of a machine trained with problems having different level of

complexity. To handle this problem, we introduce weighting into the decoding process. The weight for learner l , w_l , is calculated as follows:

$$w_l = 1/\sum_k L(D(l,k))$$
$$L(D(l,k)) = \begin{cases} 1 & if\ D(l,k) > 0 \\ 0 & else \end{cases} \quad (6)$$

where, $L(D(l,k))$ is a function for discerning positive classes from negative classes. Then, the weighted decoding is as follows:

$$s(f(x), D(\cdot,k)) = \sum_l w_l f_l(x) D(l,k) \quad (7)$$

This decoding can be used for determining the complexity of a problem. If we adopt this measure and obtain improvement in decomposition, then we can think that the decomposition generates complex problems.

3 Intuitive Problem Complexity

We define a new measure for estimating the complexity of a machine as well as the weighted decoding. We need some measure that estimates the complexity when in designing the decomposition matrix, not in the experiment as the weighted decoding.
The magnitude of a super class, equation (2), for training a binary classifier, means that how many classes are grouped into one. Intuitively, one expects that, as the number of classes that is grouped into one increases, i.e., the magnitude of $|C_l^+|$ or $|C_l^-|$ increases, the complexity of the binary problem associated with them will increase. From this viewpoint, we can say that the most complex case is $|C_l^+| = |C_l^-| = (K/2) >> 2$, and the easiest case is $|C_l^+| or |C_l^-| = 1$ when the number of classes is K . In other words, if we define intuitively the problem complexity as the magnitude of the super class of a binary problem, this can be in proportion to $|C_l^+|$ and $|C_l^-|$. Let us define Intuitive Problem Complexity (IPC) as follows:

$$IPC \equiv Min(|C^+|, |C^-|) \quad (8)$$

We summarize the magnitude of each super class of different decompositions and IPC in Table 1. According to Table 1, the tree-based scheme can be considered as a very complex problem compared to other schemes. The second complex problem can be the *N*-Shift scheme or CC scheme up to the value of *N*.

Table 1. Comparison of the magnitudes of super classes and IPC

Decomposition Scheme	OPC	All-Pairs	CC	Tree-based	*N*-Shift
$\lvert C^+ \rvert$	*1*	*1*	*2*	*K/2*	*N*
$\lvert C^- \rvert$	*K-1*	*1*	*K-2*	*K/2*	*K-N*
IPC	*1*	*1*	*2*	*K/2*	*N*

4 Experimental Results

4.1 Data sets

We used the ORL face dataset, which is one of the popular public datasets used in face recognition. The image set consists of 400 images, ten images for each individual. Each image for one person differs from each other in lighting, facial expression, and pose. We obtain the final training and testing dataset by applying preprocessing and Principal Component Analysis. Fig. 3 shows examples of the normalized face images produced after preprocessing.

Fig. 3. Some normalized facial images in the ORL dataset.

We used all of the face images for PCA transformation, and divided them into two parts; one was used for a gallery set (reference set), and the other was used for a probe set (test set). We obtained the highest recognition rate at 48-dimension with a PCA-based rank test, which is the standard test protocol described in FERET [13]. We determined the feature dimension by employing the procedure mentioned above, because the focus of our experiments is to display the classification performance of our proposed method. To compare the properties of the output coding methods, we used the SMOBR [14], which is one of the implementations of SMO [15], with RBF kernels as a base classifier. We randomly selected five images of each person, for training and the remaining five for testing. The number of samples for training and testing is both 200 respectively and the dimension of one sample is 48. Note that the dataset has a relatively small number of samples for its high dimensional feature space. We evaluated various decoding schemes on the ORL face dataset and compared their recognition performance. Table 2 shows the decoding schemes we investigated.

Table 2. Various decoding schemes.

Symbol	Meaning
HM	Hamming Decoding
MG	Margin Decoding
RD	Relative Distance Decoding
WHM	Weighted Hamming Decoding
WMG	Weighted Margin Decoding

In the subsequent section, the recognition accuracy of each decomposition scheme is presented. For those results, we calculated the recognition accuracy, varying *C* of SVM parameter from 1 to 10 and dispersion from 0.2 to 1.0 and chose the best recognition accuracy among them.

4.2 Properties Analysis

In this section, we compare and analyze empirically some properties of the representative output coding methods, such as OPC, All-Pairs, CC and our proposed OPC-based methods, on the following items.

Relationships between Overlapped Learning and Hamming Decoding: The error correcting ability is related to the minimum hamming distance of a decomposition matrix, and this is obtained from the overlapped learning of classes. We investigate it empirically. The number of binary machines generated and the minimum hamming distance of each output coding method for 40 classes are summarized in Table 3. We assume that the hamming distance between zero and zero or nonzero element of a decomposition matrix is 0.5.

Table 3. Number of machines and Minimum hamming distance of decomposition schemes.

Decomposition Scheme	Number of Machines		Minimum Hamming Distance	
	K=40	*K-Class*	*K=40*	*K-Class*
OPC	*40*	*K*	*2*	*2*
All-Pairs	*780*	*K(K-1)/2*	*390*	*(K(K-1)/2–1)/2+1*
CC	*780*	*K(K-1)/2*	*76*	*2 (K-2)*
N-Shift	*80*	*K+K*	*2*	*2*
Tree-Based	*78*	*(K-2)+K*	*2*	*2*

Fig. 4 (a) presents the recognition accuracy of each decomposition scheme with hamming decoding. If we compare the recognition accuracy of Fig. 4 (a) with the number of machines and the minimum hamming distance in Table 3, we can observe that the recognition accuracy is in proportion to both the number of machines and the minimum hamming distance.

The recognition accuracy of OPC is considerably lower than those of *N*-Shift and Tree-based schemes in spite of their having the same hamming distances. The reason for this observation is that OPC does not retain any error correction ability because it does not conduct overlapped learning. In other words, both *N*-Shift and Tree-based schemes generate some extra binary machines in addition to the same machines of the OPC scheme; as a result, this allows for them to train classes in an overlapped manner, where it makes a considerable difference. Therefore, we conclude that the recognition accuracy of each decomposition scheme with hamming decoding depends on the number of machines for overlapped learning as well as its minimum hamming distance.

Hamming Decoding versus Margin Decoding: According to Fig. 4(b), margin decoding is superior to hamming decoding for all the decomposition schemes, except for All-Pairs. This means that the margin decoding does not strongly depend on the number of machines or the minimum hamming distance. The reason for the poor accuracy of All-Pairs with margin decoding can be explained by two viewpoints as follows: First, the number of samples being used in training each machine of All-Paris is significantly smaller than that of OPC. Secondly, the decomposition matrix includes zero elements, which means that some classes exist that are not involved in training a machine. That raises the problem of nonsense outputs. The level of the nonsense outputs problem increases as the number of classes increases.

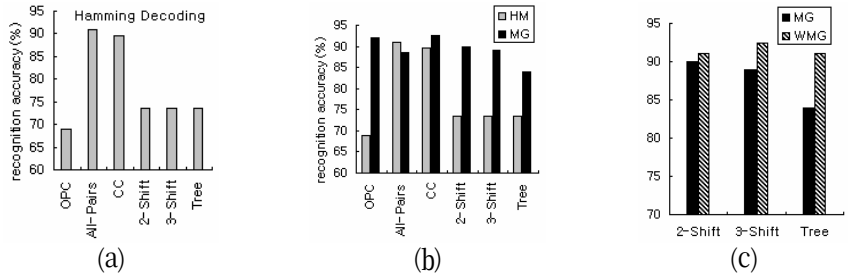

Fig. 4. Comparison of recognition accuracy (a) with hamming decoding, (b) between hamming and margin decoding, and (c) between margin and weighted margin decoding.

Relationships between Performance and Intuitive Problem Complexity: While *N*-Shift and Tree-Based schemes have more machines due to the overlapped learning, they are inferior to OPC in recognition accuracy. For explanation of the reason, we consider the Intuitive Problem Complexity (IPC) and the weighted decoding. The IPC of each decomposition scheme being computed using Table 1 with K=4, can be ordered ascendant as follow: OPC=1, 2-Shift=2, 3-Shift=3, and Tree-Based=20. This order corresponds exactly to the order of their recognition accuracy shown in Fig. 4(c).

Therefore, we infer that the overlapped learning has a strong effect when it is used with hamming decoding; however, this is not the case with margin decoding. In other words, recognition accuracy depends more on the IPC than the overlapped learning effects when we use margin decoding. Table 4 presents both the IPC and recognition accuracy on the ORL dataset.

To support this inference, We compare the recognition accuracy of *N*-Shift and Tree-Based schemes with margin decoding and weighted decoding respectively in Fig. 4(c). According to Table 4 and Fig. 4(c), the recognition accuracy of each decomposition scheme decreases as the IPC increases; however, their recognition accuracy is almost the same as our proposed weighted margin decoding. This means that weighted margin decoding can remove something related to the problem complexity represented by IPC. These results allow us to infer again that recognition accuracy strongly depends on the IPC of each decomposition matrix when we use margin decoding.

Table 4. Recognition Accuracy (RA) of decomposition schemes with margin decoding and IPC.

Decomposition	OPC	2-Shift	3-Shift	Tree-Based
RA (%)	*92.0*	*90.0*	*89.0*	*84.0*
IPC	*1*	*2*	*3*	*20*

Performance Analysis: In Table 5, we present the recognition accuracy of the experiments on the ORL dataset with various decomposition and decoding schemes.

Table 5. Rcognition accuracy (%) on the ORL face dataset.

Decomposition Scheme	Decoding Scheme				
	HM	MG	RD	WHM	WMG
OPC	69.0	92.0	**93.0**	-	-
All-Pairs	**91.0**	88.5	88.5	-	-
CC	89.5	92.5	**93.0**	-	-
2-Shift	73.5	90.0	**93.0**	73.5	91.0
3-Shift	73.5	89.0	90.0	71.5	**92.5**
Tree-Based	73.5	84.0	85.5	75.0	**91.0**

When we compare the OPC and All-Pairs, with the hamming decoding, All-Pairs shows a significantly better performance than OPC, but with the margin decoding, OPC shows a better performance. Overall, OPC with the margin decoding shows slightly better performance than All-Pairs with the hamming decoding. We infer that the performance of OPC training all classes at a time is better than that of All-Pairs since the number of training face image of one person is small.

Each machine in OPC and CC trains all the classes at a time. In this case, CC shows significantly better performance than OPC in hamming decoding like All-Pairs due to its large number of machines. With margin decoding, the performance of the two machines is almost the same regardless of their differing numbers.

Consequently, when we have small number of samples, such as face images, the OPC-like schemes training all the classes at a time can be preferred, but it is unnecessary to make too many machines for the overlapped learning like the CC scheme to improve an error correcting ability at the expense of a larger IPC than OPC and All-Pairs.

5 Conclusion

In this paper, we compared and analyzed empirically certain properties of the representative output coding methods such as OPC, All-Pairs, CC and our proposed OPC-based methods with a face recognition problem. We observed the followings: Firstly, the recognition accuracy of each decomposition scheme with a hamming decoding depends on the number of machines for overlapped learning as well as its minimum hamming distance of it. Secondly, the margin decoding is superior to hamming decoding with all the decomposition schemes except for All-Pairs. The margin decoding is slightly independent of the number of machines or the minimum hamming distance.

Thirdly, we infer that an overlapped learning can have a strong effect when it is used with the hamming decoding, but this is not the case with the margin decoding. This means that recognition accuracy relies more on the IPC than the overlapped learning effects when we use the margin decoding.

According to our experiment on face recognition, we conclude that the performance depends more on the problem complexity than the minimum hamming distance of the decomposition matrix, so it is no need to consider seriously the conventional error-correcting concept, and we suggest that the IPC of desired output coding method should be small as one.

References

1. T. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes", Journal of Artificial Intelligence Research, Vol. 2, pp. 263-286, 1995.
2. F. Masulli and G. Valentini, "Effectiveness of Error Correcting Output Codes in Multiclass Learning Problems", Proc. of the 1st Int'l Workshop on Multiple Classifier Systems, Lecture Note in Computer Science, Vol. 1857, pp. 107-15, 2000.
3. T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multi-class learning problems", Information Fusion, Vol. 4, pp. 11-21, 2003.
4. G Rassch and A. Smola, "Adapting Codes and Embeddings for Polychotomies", Advances in Neural Information Processing Systems, Vo. 15, 2003.
5. G. James and T. Hastie, "The Error Coding Method and PICTs", Computational and Graphical Statistics, Vol. 7, pp. 337-387, 1998.
6. J. Furnkranz, "Round Robin Rule Learning", Proc. of the 18th Int'l Conf. on Machine Learning, pp. 146-153, 2001.
7. V. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.
8. K. Tumar and J. Gosh, "Error Correlation and Error Reduction in Ensemble Classifier", Tech. Report, Dept. of ECE, Univ. Texas, July 11, 1996.
9. E. Allwein, R. Schapire and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers", Journal of Machine Learning Research, Vol. 1, pp. 113-141, 2000.
10. M. Moreira and E. Mayoraz, "Improved Pairwise Coupling Classification with Correcting Classifiers", Proc. of European Conf. on Machine Learning, pp. 160-171, 1998.
11. J. Ghosh, "Multiclassifier Systems: Back to the Future", Proc. of the 3rd Int'l Workshop on Multiple Classifier Systems, Lecture Note in Computer Science, Vol. 2364, pp. 1-15, 2002.
12. T. Hastie and R. Tibshirani, "Classification by Pairwise Coupling", Advances in Neural Information Processing Systems, Vol. 10, pp. 507-513, MIT Press, 1998; The Annals of Statistics, Vol. 26, No. 1, pp. 451-471, 1998.
13. P. Phillips, H. Moon, S. Rizvi and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, pp. 1090-1104, 2000.
14. M. Almedia, "SMOBR-A SMO program for training SVMs", Dept. of EE, Univ. of Minas Gerais, 2000, Available: http://www.litc.cpdee.ufmg.br /~barros/svm/ smobr.
15. J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", Tech. Report 98-14, Microsoft Research, Redmond, 1998.