Three-Dimensional Design of Error Correcting Output Codes

Mohammad Ali Bagheri[1], Qigang Gao[1], and Sergio Escalera[2]

[1] Faculty of Computer Science, Dalhousie University, Halifax, Canada
[2] Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
Dept. Matemtica Aplicada i Anlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain

Abstract. Error-correcting output codes (ECOC) represent a powerful framework to deal with multi-class classification problems based on the combination of binary classifiers. The key factor affecting the performance of ECOC methods is the independence of binary classifiers, without which the ECOC method would be ineffective. In this paper, we propose an efficient new approach to the classical ECOC design in order to improve independency among classifiers. The underlying rationale for our work is that we design three-dimensional codematrix, where the third dimension is the feature space of the problem domain. In addition to creating more independent classifiers, ECOC matrices with longer codes can be built. This paper provides a set of experimental results on 12 datasets using three base learners: Neural Networks, Decision Tree, and AdaBoost. The results show that the proposed technique increases the classification accuracy in comparison with the state of the art ECOC coding methods.

Keywords: Error correcting output codes, multiclass classification, feature subspace, ensemble classification

1 Introduction

A common task in many real world pattern recognition applications is to discriminate between instances that belong to multiple classes. In contrast to this, most of the established classification algorithms, such as Support Vector Machine (SVM) [3] and Multi-Layer Perceptron (MLP), work better facing two-class problems. The predominant approach to overcome this problem is to recast the multi-class problem into a series of smaller binary classification tasks, which is referred to as "class binarization" [13]. In this way, two-class problems can be solved by binary classifiers and the results can then be combined so as to provide a solution to the original multiclass problem. Among the proposed methods for approaching class binarization, three widely applied strategies are one-versus-one [15], one-versus-all [6] [2], and Error Correcting Output Codes (ECOC) [9].

[3] Indeed, the SVM algorithm is specifically designed for problems with only two target classes

In one-versus-all, the multiclass problem is decomposed into several binary problems in that for each class a binary classifier is trained to discriminate among the patterns of the class and the patterns of the remaining classes. In the one-versus-one approach, one classifier is trained for each possible pair of classes. In both approaches, the final classification prediction is based on a voting or committee procedure. On the other hand, Dietterich and Bakiri [9] presented a general framework for class binarization approaches in order to enhance generalization ability of binary classifiers, which is known as Error Correcting Output Codes (ECOC).

The basis of the ECOC framework is it to decompose a multiclass problem into a larger number of binary problems. In this way, each classifier is trained on a two meta-class problem, where each meta-class consists of some combinations of the original classes. The ECOC method can be broken down into two stages: encoding and decoding. The aim of the encoding stage is to design a discrete decomposition matrix (codematrix) for the given problem. Each row of the codematrix, called codeword, is a sequence of bits representing each class, where each bit identifies the membership of the class to a classifier [11]. In the decoding stage, the final classification decision is obtained based on the outputs of binary classifiers. Given an unlabeled test sample, each binary classifier casts a vote to one of the two meta-classes used in training. This vector is compared to each class codeword of the matrix, and the test sample is assigned to the class with the closest codeword according to some distance measure. Because of its ability to correct the bias and variance errors of the base classifiers [20], [21], [27], the ECOC framework has been successfully applied to a wide range of applications [7], [10], [3].

However, the extensive results of [21], [13] show that the success of the ECOC approach strongly depends on the independency of the binary classifiers, a term which is known as classifier diversity in ensemble classification literature. There exist some previous studies that have proposed the use of bagging and boosting within the ECOC method, mainly by selecting a sampling of data for each dichotomizer, in order to increase the diversity of the binary problems. In this sense, Schapire proposed a new technique by combining boosting algorithm with the idea of output codes [24]. Similarly, Windeatt and Ardeshir proposed to combine AdaBoost with output coding using decision tree as a base learner [26]. Although performing sampling of data, previous methods use the same set of available features, so it is likely that some classification errors will be common, arising from noisy or non-discriminant features. One efficient approach to increase diversity among an ensemble of classifiers is to train each learner with data that consist of different feature subsets, leading to uncorrelated errors of base learners [16], [28]. This idea, usually called subspace approach, can effectively make use of diversity of base learners to reduce the variance as well as the bias errors [14], [25]. Inspired by this idea, we design a new strategy for the ECOC approach. The proposed strategy uses different feature subsets for each dichotomizer, leading to more diverse classifiers and, therefore, increasing the overall system accuracy. In addition to the design of more independent classifiers, it also allows for the design of larger codes in comparison to classical approaches.

The rest of this paper is organized as follows: Section 2 briefly reviews the three main class binarization methods. The proposed method for binary classifier selection is explained in detail in Section 3. Section 4 reports and analyses the experimental results. Finally, Section 5 concludes the paper.

2 Related work

The following briefly describes some notations used in this paper:

- $T = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_m}, y_m)\}$. A training set; where $\mathbf{x_i} \in R^n$; and each label,y_i, is an integer belongs to $Y = \{\omega_1, \omega_2, \ldots, \omega_c\}$, where c is the number of classes
- $h = \{h_1, h_2, \ldots, h_L\}$: A set of L binary classifiers.

The goal of class binarization methods it to get a feature vector,$\mathbf{x}$, as its input, and to assign it to a class label from Y. As we mentioned before, the methods for multiclass problems can be generally categorized into three approaches:
One-versus-all(OVA): The one-versus-all method constructs c binary classifiers, one for each class. The *ith* classifier, h_i, is trained with data from class i as positive instances and all data from the other classes as negative instances. A new instance is classified in the class whose corresponding classifier output has the largest value.
One-versus-one (OVO): The one-vs-one method, also called pairwise classification, constructs $c(c-1)/2$ classifiers [19]. Classifier h_{ij} is trained using all data from class i as positive instances and all data from class j as negative instances, and disregarding the remaining data. To classify a new instance, $\mathbf{x}$, each of the base classifiers cast a vote for one of the two classes used in its training. Then, the one-versus-one method applies the majority voting scheme for labeling $\mathbf{x}$ to the class with the most votes. Ties are usually broken arbitrarily for the larger class. More complicated combination methods have also been proposed [22] [4].

Error Correcting Output Codes (ECOC): The basis of the ECOC framework consists of designing a codeword for each of the classes. This method uses a matrix M of $\{1, -1\}$ values of size $c \times L$, where L is the number of codewords codifying each class. This matrix is interpreted as a set of L binary learning problems, one for each column. That is, each column corresponds to a binary classifier, called *dichotomizer* h_j, which separates the set of classes into two metaclasses. Instance $\mathbf{x}$, belonging to class i, is a positive instance for the jth classifier if and only if $M_{ij} = 1$ and is a negative instance if and only if $M_{ij} = -1$.

When testing an unlabeled pattern, $\mathbf{x}$, each classifier outputs a "0" or "1", creating a L long output code vector. This output vector is compared to each codeword in the matrix, and the class whose codeword has the closest distance to the output vector is chosen as the predicted class. The process of merging the outputs of individual binary classifiers is usually called decoding. The most common method for decoding is the Hamming distance.

The ECOC method was then extended by Allwein et al. [1] using a coding matrix with three values, $\{1, 0, -1\}$, where the zero value means that a given class is not considered in the training phase of a particular classifier. In this way, a class can be omitted in the training of a particular binary classifier. This extended codeword is denominated sparse random code and the standard codes (binary ECOC) were named dense random codes. Thanks to this unifying approach, the classical one-versus-one method can be represented as an ECOC matrix: the coding matrix has

3 Three-dimensional coding design

As we mentioned before, one of the key factors to the success of ECOC methods is the independence of binary classifiers, without which the output coding approach would be ineffective [20], [21]. In ECOC methods, the codematrix can be considered as the core component to generate independent classifiers. Accordingly, most previous methods to design the ECOC matrix try to build an optimal codematrix, usually by optimizing the row separation and column separation criteria. Many researchers, however, agree that random generation of codematrix is a "reasonably" good method and that "more sophisticated methods might have only marginal effect on testing error" [9], [24], [13]. It has also shown that large random codes would not be outperformed by codes designed for their error-correcting capabilities [18]. Therefore, the overall performance of ECOC codes built by different strategies for a same base classifier tends to be very similar, especially as the length of codewords increases.

Inspired by the subspace approach in ensemble learning, we propose a new method to improve the efficiency of the ECOC approach. The proposed approach is based on considering the feature space in the design process of the ECOC matrix. That is, each dichotomizer is trained with a different feature subset, leading to better classification accuracy. From the design process point of view, we generate three-dimensional codematrix, where the third dimension is the feature space of the problem domain. To generate this framework, first a two-dimensional codematrix is generated from a previous set of matrices that maximizes the minimum distances between any pair of codewords. Then, for each column, a random vector of $-1, +1$ values of size n is generated; where n is the number of features. The meaning of '+1' ('-1') in the vector is that the corresponding feature is (not) included in the corresponding classifier. Note that in both, sparse and dense coding styles, the value of each cell in the feature space cannot take the 0 value. The representation of the proposed three-dimensional codematrix is illustrated in Fig. 1.

This approach, not only increases the independence among classifiers, but it can also build longer codewords. It can be shown that the maximum length of codewords in ECOC matrices is $(2^{c-1} - 1)$ and $(3^c - 2^{c+1} + 1)/2$ for dense and sparse coding styles, respectively. Thus, the maximum number of binary classi-

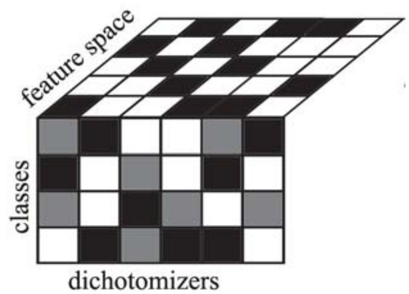


Fig. 1. The Subspace ECOC approach

fiers in the classical ECOC methods is small in problems with relatively small number of classes ($c < 6$). It should be noted that for a three-class problem, the dense ECOC matrix is equivalent to the one-versus-all ECOC matrix. Conversely, the ECOC method with longer codes is able to significantly improve the results [1], [13]. In our proposed approach, since each classifier can be trained using a variety of feature subsets, a larger set of classifiers can be built. In an n-dimensional feature space, $2^n - 1$ different non-empty feature subsets can be selected. So, the number of distinct dichotomizers is $(2^n - 1).(2^{c-1} - 1)$ and $(2^n - 1).(3^c - 2^{c+1} + 1)/2$ for dense and sparse ECOC, respectively. The other advantage of the subspace approach is that each binary classifier requires less training time, since it uses fewer features.

4 Experimental comparison

4.1 Experimental settings

In order to investigate the relative performance of our proposed method, an empirical study was conducted. We compared our proposed method with OVO, OVA, and classical dense and sparse ECOC methods on 12 multiclass datasets from the UCI machine learning repository [5], which are summarized in Table 1. The class of an instance in the ECOC schemes is chosen using the Exponential Loss-Weighted (ELW) decoding [12]. The length of codewords (number of nontrivial dichotomies) is also shown for OVO, dense and sparse ECOC methods. We considered random codes of $10log_2(c)$ and $15log_2(c)$ bits for dense and sparse ECOC, respectively [1]. To limit the computational complexity of the experiments, we set the number of different feature subsets per each nontrivial dichotomizer as 10. Thus, codewords are 10 times longer in 3d-ECOC design for both dense and sparse ECOC methods.

In this study, three base learners were chosen: multilayer perceptron (MLP) with 10 hidden nodes and the hyperbolic tangent transfer function, Gentle AdaBoost with 50 runs of decision stumps, and a classification and regression tree (CART) with the Gini-index as a split criterion.

Table 1. Summary of the used datasets

Dataset	**# instances**	**# features**	**# classes**	length of codewords		
				OVO	dense ECOC	sparse ECOC
Abalone	4177	8	3	3	16	24
Cmc	1473	9	3	3	16	24
Derm	366	34	6	15	26	39
Ecoli	336	7	8	28	30	45
Glass	214	10	7	21	28	42
Lymph	148	18	4	6	20	30
Optdigits	5620	64	10	45	33	50
Sat	6435	36	6	15	26	39
vehicle	846	18	3	3	16	24
Wine	178	13	3	3	16	24
Yeast	1484	8	10	45	33	50
Zoo	101	16	7	21	28	42

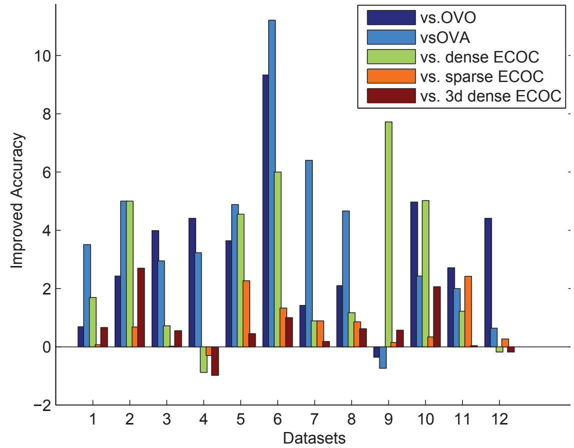
The experiments were all implemented in MATLAB software. For performance evaluation, we utilized 10-fold cross-validation to improve the reliability of the results. In order to have a fair comparison, the training and test sets of all methods were the same for each repetition of the experiments.

4.2 Experimental results

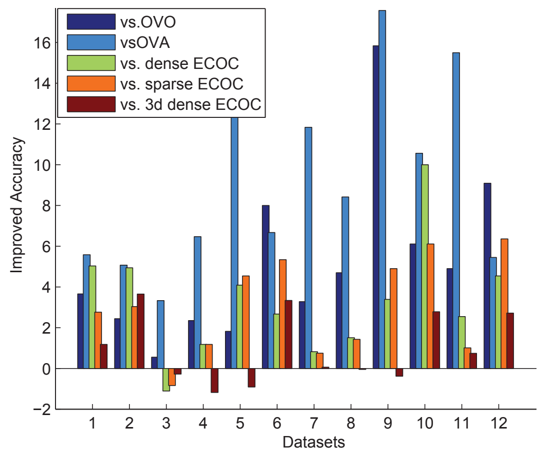
The average accuracy of the considered methods for the 12 datasets is presented in Table 2 -Table 4. In theses tables, the means of prediction accuracy over 10 runs (expressed in %) are reported for each classification method on the considered datasets. The comparison of the methods is also illustrated in Fig. 2. In this figure, the improved accuracy using 3D-sparse ECOC design versus other methods on all considered datasets is presented.

In order to see whether the proposed method is significantly better or worse than other methods, statistical analysis is necessary. According to the recommendations of Demsar [8], we consider the use of non-parametric tests. Non-parametric tests are safer than parametric tests, such as ANOVA and t-test, since they do not assume normal distribution or homogeneity of variance. In this study, we employ the Iman-Davenport test. If there are statistically significant differences in the classification performance, then we can proceed with the Nemenyi test as a post hoc test, which is used to compare six methods with each other.

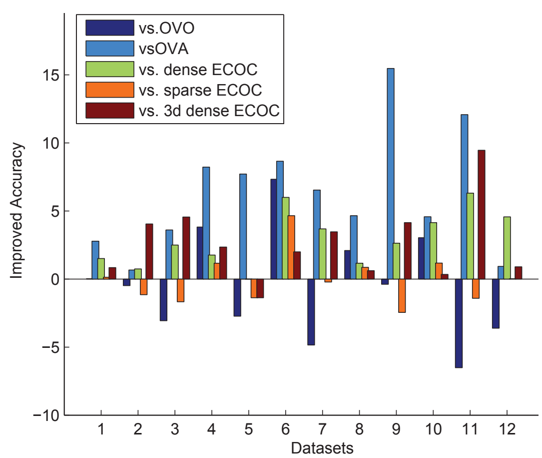
We first rank competing methods for each dataset. The best performing method getting the rank of 1, the second best ranked 2, so on and so forth. A method's mean rank is obtained by averaging its ranks across all datasets. Then, we use the Friedman test [8] to compare these mean ranks to decide whether to reject the null hypothesis, which states that all considered methods have equivalent performance. Iman and Davenport [17] found that this statistic is undesirably conservative, and proposed a corrected measure. Applying this



(a) MLP



(b) CART



(c) AdaBoost

Fig. 2. Improved classification accuracy using 3D-sparse ECOC design versus other methods (a) (b) and (c)

method, we can reject the null hypothesis, that is, there exists significant statistical difference among the rival methods.

Further, to compare rival methods with each other, we apply the Nemenyi test, as illustrated in Fig. 3. In this figure, the mean rank of each method is indicated by a square. The horizontal bar across each square shows the critical difference. Two methods are significantly different if their corresponding average ranks differ by at least the critical difference value. That is, their horizontal bars are not overlapping.

Table 2. Classification accuracies of different methods using MLP

	OVO	OVA	Dense ECOC	Sparse ECOC	3D dense ECOC	3D sparse ECOC
Abalone	66.17	63.35	65.17	66.79	66.20	66.86
cmc	52.57	50.00	50.00	54.32	52.30	55.00
derm	93.63	94.67	96.90	97.60	97.07	97.62
ecoli	85.00	86.18	90.29	89.71	90.39	89.41
glass	61.36	60.12	60.45	62.73	64.55	65.00
lymph	78.00	76.12	81.33	86.00	86.33	87.33
optdigits	97.51	92.53	98.04	98.04	98.75	98.93
sat	88.12	85.56	89.05	89.36	89.60	90.22
vehicle	81.73	82.11	73.65	81.22	80.80	81.37
wine	93.74	96.28	93.69	98.37	96.65	98.71
yeast	58.06	58.77	59.55	58.35	60.73	60.77
zoo	90.18	93.95	94.77	94.32	94.77	94.59
Mean accuracy	78.84	78.30	79.41	81.40	81.51	82.15

4.3 Experimental result analysis

As can be seen in above tables, the proposed approach is generally able to outperform all the other methods for the three types of base learners. As a general conclusion, the advanced performance of the method does not differ much depending on the base classifier. An analysis of the results shows a somewhat clearer picture:

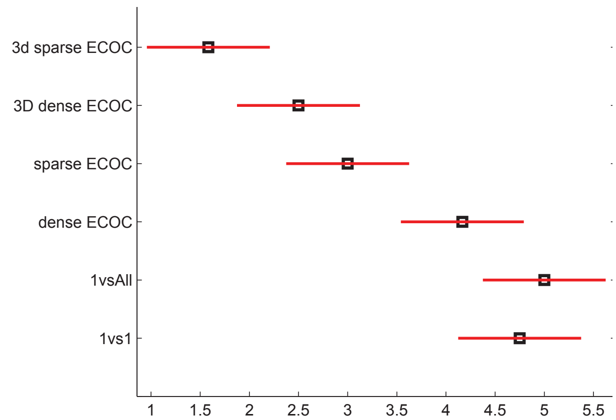
Result analysis using MLP and Decision Tree: For both MLP and Decision Tree as the base learner, we found significant differences between Subspace ECOC and classical ECOC for both dense and sparse schemes. As can be seen in Table 3 - Table 8, the proposed method is generally able to outperform the rest of methods. It is also ranked as the preferred method in these classification algorithms. An analysis of the results shows that when the number of training patterns is relatively small compared with the dimensionality of data, the subspace approach is usually a better choice. In these cases, the training sample size relatively increases. Ho [16] showed that while most classification approaches suffer from the curse of dimensionality, the subspace approach can take advantage of high dimensionality of data.

Table 3. Classification accuracies of different methods using DT

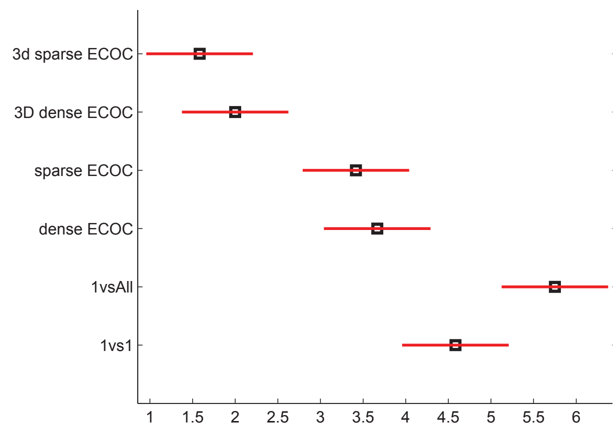
	OVO	OVA	Dense ECOC	Sparse ECOC	3D dense ECOC	3D sparse ECOC
abalone	59.67	57.75	58.30	60.57	62.15	63.33
cmc	51.55	48.92	49.05	50.95	50.34	53.99
derm	96.67	93.89	98.33	98.06	97.50	97.22
ecoli	84.12	80.00	85.29	85.29	87.65	86.47
glass	70.45	58.18	68.18	67.73	73.18	72.27
lymph	76.67	78.00	82.00	79.33	81.33	84.67
optdigits	94.98	86.42	97.44	97.51	98.20	98.26
sat	87.52	83.80	90.71	90.79	92.27	92.22
vehicle	72.87	68.84	73.34	75.04	76.78	77.18
wine	91.67	87.22	87.78	91.67	95.00	97.78
yeast	55.57	44.97	57.92	59.46	59.73	60.47
zoo	88.18	91.82	92.73	90.91	94.55	97.27
Mean accuracy	77.49	73.32	78.42	78.94	80.72	81.76

Table 4. Classification accuracies of different methods using AdaBoost

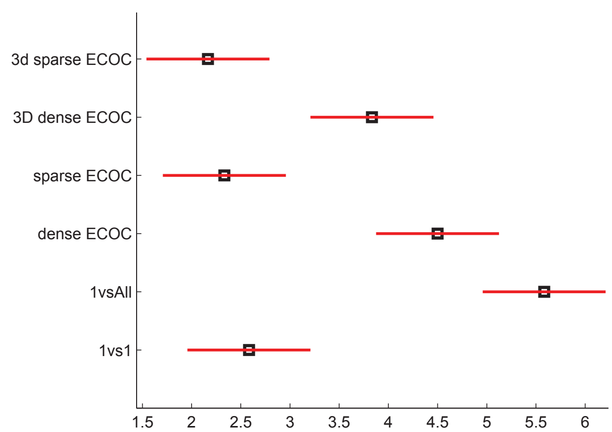
	OVO	OVA	Dense ECOC	Sparse ECOC	3D dense ECOC	3D sparse ECOC
abalone	64.98	62.22	63.49	64.86	64.16	65.00
cmc	55.41	54.26	54.19	56.08	50.88	54.93
derm	93.06	86.39	87.50	91.67	85.44	90.00
ecoli	81.47	77.06	83.53	84.12	82.94	85.29
glass	75.00	64.55	72.27	73.64	73.64	72.27
lymph	78.00	76.67	79.33	80.67	83.33	85.33
optdigits	96.71	85.32	88.17	92.08	88.39	91.86
sat	87.12	84.56	88.05	88.36	88.60	89.22
vehicle	77.88	72.24	78.47	79.65	77.41	78.71
wine	94.25	92.71	93.14	96.11	96.95	97.29
yeast	59.06	40.47	46.24	53.96	43.09	52.55
zoo	95.45	90.91	87.27	91.82	90.94	91.84
Mean accuracy	79.87	73.95	76.80	79.42	77.15	79.52



(a) MLP



(b) CART



(c) AdaBoost

Fig. 3. Comparison results of rival methods using the Nemenyi test (a) (b) and (c)

Comparing dense ECOC and sparse ECOC and also Subspace dense ECOC and Subspace sparse ECOC, we can see that sparse coding design achieved an improved accuracy results. The main reason behind this improvement is that in the sparse coding design we can generate longer codewords and it has been shown that the ECOC method with longer codes is able to significantly improve the results [1], [13]. However, this improvement is not significant in terms of statistical test.

Result analysis using AdaBoost: The case for AdaBoost is different. Using AdaBoost as the base learner, we can see that Subspace sparse ECOC, sparse ECOC, and one-versus-one methods achieve the better accuracy results. These interesting results demonstrate that AdaBoost algorithm works better using ternary coding design. This finding is similar to the behavior of SVMs classifiers; i.e. it has been shown that SVM classifiers work better using ternary coding, especially one-versus-one. The remarkable issue is that both AdaBoost and SVM algorithms are conceptually similar, in which both algorithms maximize the margin between instances of different classes [23].

5 Conclusion

In this paper, we presented a novel three-dimensional approach of Error-Correcting Output Codes to deal with multi-class classification problems. The proposed technique is based on designing the ECOC matrix code using different random subspace in order to generate more independent classifiers. For this task, each dichotomizer is trained using different feature subset, and the coding matrix is recoded using the L-norm distance. In addition to creating more independent classifiers in the proposed Subspace ECOC technique, ECOC matrices with longer codes can be generated. The experimental evaluation over several UCI Machine Learning repository datasets shows that significant performance improvements can be obtained compared to the one-versus-one, one-versus-all, and classical dense and sparse ECOC methods.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2001)
2. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. Neural Networks, IEEE Transactions on 6(1), 117–124 (1995)
3. Bagheri, M., Escalera, S.: Error correcting output codes for multiclass classification: Application to two image vision problems. In: 16th International symposium on Artificial Intelligence and Signal Processing. p. to appear. Shiraz, Iran (2012)
4. Bagheri, M., Gao, Q., Escalera, S.: Efficient pairwise classification using local cross off strategy. In: 25th Canadian conf. on Artificial Intelligence. p. to appear. Toronto, Canada (2012)

5. Blake, C., Merz, C.: Uci repository of machine learning databases, department of information and computer sciences, university of california, irvine (1998)
6. Clark, P., Boswell, R.: Rule induction with cn2: Some recent improvements. In: Kodratoff, Y. (ed.) Machine Learning EWSL-91, Lecture Notes in Computer Science, vol. 482, pp. 151–163. Springer Berlin / Heidelberg (1991)
7. David, A., Lerner, B.: Support vector machine-based image classification for genetic syndrome diagnosis. Pattern Recognition Letters 26(8), 1029–1038 (2005)
8. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
9. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263286 (1995)
10. Elif Derya, b.: Multiclass support vector machines for diagnosis of erythemato-squamous diseases. Expert Systems with Applications 35(4), 1733–1740 (2008)
11. Escalera, S., Pujol, O., Radeva, P.: Separability of ternary codes for sparse designs of error-correcting output codes. Pattern Recognition Letters 30(3), 285–297 (2009)
12. Escalera, S., Pujol, O., Radeva, P.: Re-coding ecocs without re-training. Pattern Recognition Letters 31, 555562 (2010)
13. Garcia-Pedrajas, N., Ortiz-Boyer, D.: An empirical study of binary classifier fusion methods for multiclass classification. Information Fusion 12(2), 111–130 (2011)
14. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Comput 4, 1–58 (1995)
15. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. In: Proceedings of the 1997 conference on Advances in neural information processing systems 10. pp. 507–513. NIPS '97, MIT Press, Cambridge, MA, USA (1998)
16. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattem Analysis and Machine Intelligence 20, 832–844 (1998)
17. Iman, R., Davenport, J.: Approximations of the critical regions of the friedman statistic. Communications in Statistics 6, 571–595 (1980)
18. James, G.M., Hastie, T.: The error coding method and pict's. Computational and Graphical Statistics 7, 377–387 (1998)
19. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Fogelman, J. (ed.) Neurocomputing: Algorithms, Architectures and Applications. Springer-Verlag (1990)
20. Kong, E., Dietterich, T.: Error-correcting output coding corrects bias and variance. In: Prieditis, A., Lemmer, J. (eds.) Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning, pp. 313–321 (1995)
21. Kong, E., Dietterich, T.: Why error-correcting output coding works with decision trees. Tech. rep., Technical Report, Department of Computer Science, Oregon State University, Corvallis, OR. (1995)
22. Park, S.H., Fürnkranz, J.: Efficient pairwise classification. In: Kok, J.N., Koronacki, J., López de Mántaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) Proceedings of the 18th European Conference on Machine Learning (ECML 2007, Warsaw, Poland). pp. 658–665. Springer-Verlag (2007)
23. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6(3), 21–45 (2006)
24. Schapire, R.: Using output codes to boost multiclass learning problems. In: Proceedings of the 14th International Conference on Machine Learning. pp. 313–321. San Francisco, CA, Morgan Kauffman, Los Altos, CA, (1997)
25. Tumer, K., Ghosh, J.: Linear and order statistics combiners for pattern classification. In: Sharkey, A. (ed.) Combining Artificial Neural Nets, pp. 127–162. Springer, Berlin (1999)

26. Windeatt, T., Ardeshir, G.: Boosted ecoc ensembles for face recognition (2003)
27. Windeatt, T., Ghaderi, R.: Coding and decoding strategies for multi-class learning problems. Information Fusion 4(1), 11–21 (2003)
28. Zor, C., Windeatt, T., Yanikoglu, B.: Bias-variance analysis of ecoc and bagging using neural nets. Studies in Computational Intelligence 373, 59–73 (2011)