Error Correcting Output Codes for multiclass classification: Application to two image vision problems

Mohammad ali Bagheri*† Gholam Ali Montazer *‡ Sergio Escalera§¶
* Department of Information Technology, School of Engineering, Tarbiat Modares University, Tehran, Iran
§ Centre de Visio per Computador, Campus UAB, Edifici O, Bellaterra, 08193 Barcelona, Spain
Email: †a.bagheri@modares.ac.ir, ‡ montazer@modares.ac.ir ¶sergio@maia.ub.es

***Abstract*—Error-correcting output codes (ECOC) represents a powerful framework to deal with multiclass classification problems based on combining 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 spite of its ability on classification of problems with relatively large number of classes, it has been applied in few real world problems. In this paper, we investigate the behavior of the ECOC approach on two image vision problems: logo recognition and shape classification using Decision Tree and AdaBoost as the base learners. The results show that the ECOC method can be used to improve the classification performance in comparison with the classical multiclass approaches.**

***Index Terms*—Error Correcting Output Codes (ECOC), logo recognition, shape categorization, multiclass classification, one-versus-one, one-versus-all.**

I. Introduction

A common task in many real-world pattern recognition problems is to discriminate among instances that belong to multiple classes. The predominant approach to deal with such problems is to recast the multiclass problem into a series of smaller binary classification tasks, which is referred to as "class binarization" [1]. 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 [2], one-versus-all [3] [4], and Error Correcting Output Codes (ECOC) [5]. 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, ECOC is a general framework for class binarization approaches that enhances the generalization ability of binary classifiers [5].

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 [6]. 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. Moreover, the ECOC ensemble has shown to reduce the bias and variance errors of the base classifiers [7], [8], [9],

In terms of classification performance, Hsu and Lin [10] compared different approaches for multiclass SVM problems, including one-versus-one, one-versus-all, and DDAG. Using ten benchmark datasets, the authors claimed that the one-versus-one method is superior to the other approaches. However, the experiments are not conclusive as only 10 datasets are used, and only two of them have more than seven classes. Furthermore, only one classification algorithm is considered in their experiments. Pedrajas and Boyer's prominent paper [1] later presented an in-depth critical assessment of the three basic multiclass methods. One of the main paper's conclusions states that ECOC and one-versus-one are the best choices for powerful learners and for simpler learners, respectively.

In the spite of ability of the ECOC approach on classification of problems with relatively large number of classes, one-versus-one and one-versus-all methods are the two most commonly used in real world applications, mainly because of their clarity in comparison with the ECOC approach. In this paper, we have applied the ECOC approach on two image processing problems: logo recognition and shape classification using two base learners. The results show that the ECOC method can be used to improve the classification performance in comparison with classical multiclass approaches.

The rest of this paper is organized as follows: Section 2 briefly reviews the three main class binarization methods. Section 3 includes the description of our applications together with experimental results and the discussion. Finally, Section 4 concludes the paper.

978-1-4673-1479-4/12/$31.00 ©2012 IEEE

II. 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,**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. So, the ensemble decision function, h, is defined as:

$$y = \arg \max_{i \in \{1,\ldots,c\}} h_i(x) \tag{1}$$

One-versus-one (OVO): The one-vs-one method, also called pairwise classification, constructs $c(c-1)/2$ classifiers [11]. Classifier ij, 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, **x**, each of the base classifiers cast a vote for one of the two classes used in its training. Then, the one-vs.-one method applies the majority voting scheme for labeling **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 [12] [13] [14].

Ko and Byun proposed a method based on combination of the one-versus-all method and a modification of the one-versus-one method using SVM as a base learner [15]. This method first obtain the top two classes whose corresponding classifiers have the highest confident based on the outputs of all one-versus-all classifiers. In a recent paper [16], very similar idea is presented and named A&O. However, in both these methods, the learning algorithm which finds the two most likely classes is the same as the final classification algorithm. Consequently, it is very likely that some classification errors will be common, arising from the limitation of base learner on certain patterns.

Error Correcting Output Codes (ECOC):The basis of the ECOC framework consists of designing a codeword for each of the classes [5]. 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 **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 $M_{ij} = -1$ [5]. Table 1 shows a possible binary coding matrix for a 4-class problem $\omega_1, \ldots, \omega_4$ with respective codewords that uses 6 dichotomizers $h_1, , h_6$. In this table, each column is associated with a dichotomy classifier,h_j, and each row is a unique codeword that is associated with an individual target class. For example, h_3 recognizes two meta-classes: original classes 1 and 4 form the first meta-class, and the other two form the second one.

TABLE I
A SAMPLE ECOC MATRIX

Class	h_1	h_2	h_3	h_4	h_5	h_6
ω_1	1	-1	1	-1	-1	1
ω_2	1	1	-1	-1	1	-1
ω_3	-1	1	-1	1	-1	1
ω_4	-1	-1	1	-1	1	1

When testing an unlabeled pattern, **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 commonly decoding methods are the Hamming distance. This method looks for the minimum distance between the prediction vector and codewords.

The ECOC method was then extended by Allwein et al. [17] 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 $\binom{n}{k}$ columns, in which each column corresponds to a pair of classes (ω_i, ω_j). For this column, the matrix has +1 in row i, -1 in row j, and zeros in all other rows. Note that in both dense and sparse coding styles, a test codeword cannot contain the zero value since the output of each dichotomizer is $-1, +1$.

III. Experimental comparison

A. Experimental settings

In order to present the results, first, we discuss the experimental settings of the experiments .We compared the Dense and Sparse ECOC method designs with classical multiclass classification methods including OVO and OVA on two real world machine vision applications. We considered random codes of $10log_2(c)$ and $15log_2(c)$ bits for dense and sparse ECOC, respectively [17]. The class of an instance in the ECOC schemes is chosen using the Exponential Loss-Weighted (ELW) decoding [16].

In this study, two base learners were chosen: Gentle AdaBoost with 50 runs of decision stumps [20], and a classification and regression tree (CART) with the Gini-index as a split criterion. 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

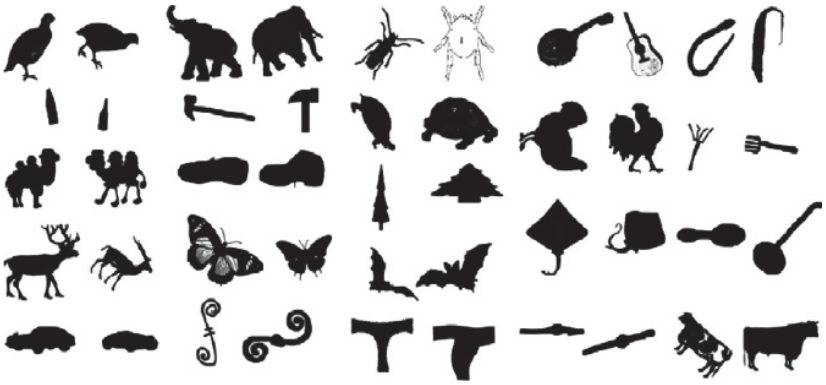


Fig. 1. Some examples of labeled shapes in the MPEG7 dataset

order to have a fair comparison, the training and test sets of all methods were the same for each repetition of the experiments.

B. Image vision applications

1) Shape categorization: The first real application on our experiments was shape classification, in which we used the MPEG7 dataset [1]. This dataset consists of $C = 70$ classes (bone, chicken, cellular phone, etc.) with 20 instances per class, which represents a total of 1400 object images. All samples were described using the Blurred Shape Model descriptor [18]. Figure 1 shows a couple of samples for some categories of this dataset.

2) Logo Recognition: The proposed approach was then used in our logo recognition problem. The logo images were based on a database of logos which contains pure pictorial logos (e.g. logo 60, Fig. 2), text-like logos (e.g. logo 30, Fig. 2), and text-graphics mixture logos (e.g. logo10, Fig. 3). The complete dataset contains 105 images and was obtained from the database distributed by the Document Processing Group, Center for Automation Research, University of Maryland. The logos in the dataset have very different sizes; the smallest one is 121 $\times$145 pixels and the largest one is 802 $\times$ 228 pixels.

This dataset provides only a single instance of 105 individual logo classes. For each logo class, some artificially degraded images were generated by using the noise models described in the following subsection. The recognition process should be translation, scale and rotation invariant. Thus, three normalization steps were applied to the raw images before feature extraction step. First, we shifted the image in order to locating the centroid of the white pixel locations at the image center, which gives us translational invariance. Second, we rotate the image around the centroid so that its major principal axis is aligned with the horizontal, which gives us rotational invariance. Third, we resize the image so that the bounding box of the logo symbol is fixed at 250 $\times$ 250 pixels.

Noise models

We investigated the robustness of the methods when the logos are corrupted using two different image degradation methods: 1) Gaussian noise (a global degradation as shown in Fig.3a) and 2) spot noise (a local degradation as shown in Fig. 3b). For each method, we degraded the images in the database, varying the amount of degradation in equally spaced steps.

We generated a set of 40 examples for each class of logo images by adding both Gaussian white noise of $mean = [0, 0.1, 0.2, \ldots, 0.5]$ and $var = [0, 0.01, \ldots, 0.05]$ and spot noise of different size $(width = 10, 15, \ldots, 30 pixels)$.

Feature extraction

Generally, there are two approaches in shape description and feature extraction: contour-based vs. region-based [19]. Region based methods considers all the pixels within a shape region to obtain the shape features, rather than only use boundary information as in contour-based techniques. Region-based methods are generally less sensitive to noise and variations; are more robust as they use all the shape information available; provide more accurate retrieval; and can cope well with shape defection, which is a common problem for contour-based shape representation methods. Especially, shape content is more important than the contour features in our application. In this paper, we used one of the commonly used region-based shape descriptor: Geometric moment invariants, which can be defined as follows:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) p, q = 0, 1, 2, \ldots \qquad (2)$$

In our experiments, we derived all combinations of lower order moments $(p, q = 0, 1, 2, 3)$, which has the desirable properties of being invariant under scaling, translation and rotation.

C. Experimental results and analysis

The average accuracy of all considered methods over 20 runs is illustrated in Figure 4 and Fig. 5 using MPEG7 and Logo datasets, respectively. These figures show the improved ability of the ECOC method in terms of classification accuracy.

In order to show the superiority of the ECOC method, statistical analysis is necessary. According to the recommendations of Demsar [20], 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 [21]. If there are statistically

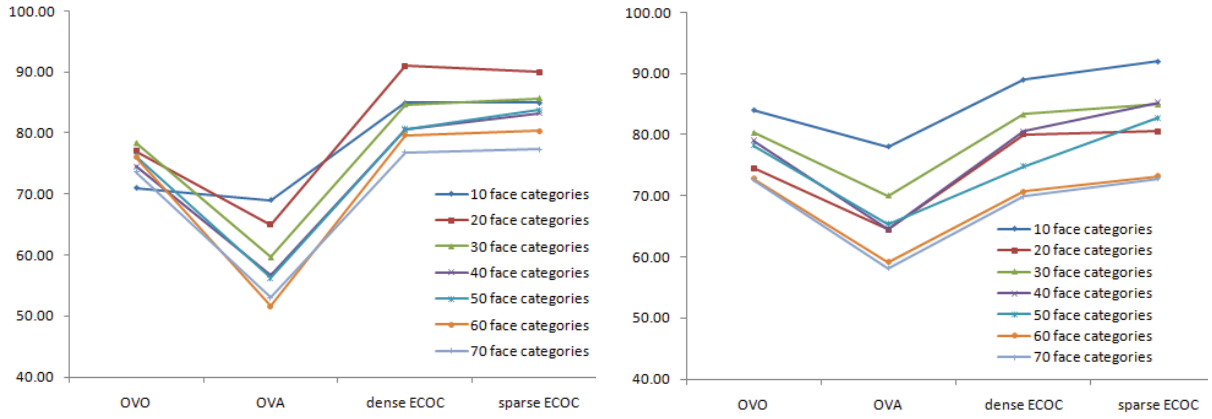
[1]MPEG7 Repository dataset: http://www.cis.temple.edu/ latecki/

(a) Logo60 (b) Logo30 (c) Logo10

Fig. 2. Some examples of logos in the database used in our experiments

(a) Gaussian noise (b) spot noise

Fig. 3. Examples of noisy logos patterns derived by applying the Gaussian and spot noise model



(a) CART Classifier (b) AdaBoost Classifier

Fig. 4. Average Accuracy of different methods using MPEG7 dataset

significant differences in the classification performance, then we can proceed with the Nemenyi test [20] as a post-hoc test, which is used to compare all methods with each other.

To do that, we first rank competing methods for each dataset. The best performing method getting the rank of 1, the second best ranked 2, etc. The method's mean rank is obtained by averaging its ranks across all experiments. Then, we use the Friedman test [18] 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 [21] found that this statistic is undesirably conservative, and proposed a corrected measure. Applying this method, we can reject the null hypothesis, and show that there exists significant statistical difference among the rival methods.

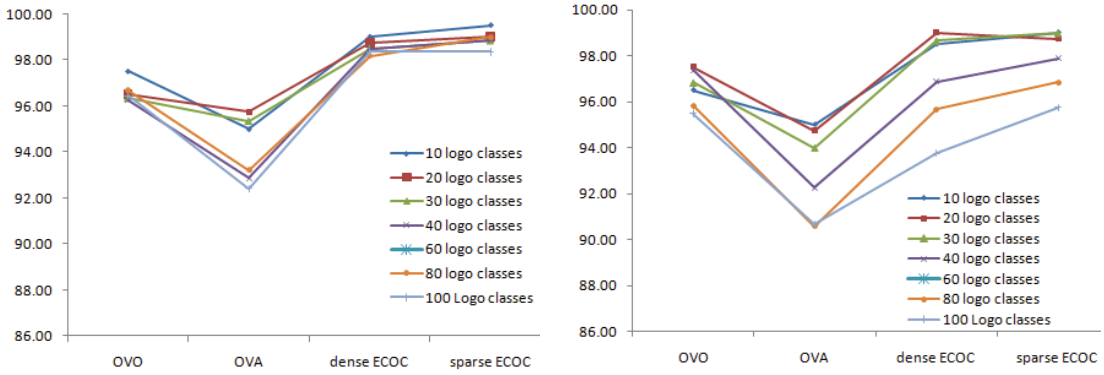
Further, to compare rival methods with each other, we apply the Nemenyi test, as illustrated in Fig. 6 and Fig.7. In these figures, the mean rank of each method is indicated by a square. The horizontal bar across each square shows the critical difference. Two methods are significantly different if their corresponding average ranks differ by at least the critical difference value. i.e., their horizontal bars are not overlapping.

The results in Fig. 4 and Fig. 5, along with the statistical tests presented in Fig. 6 and Fig. 7 indicate that in general ECOC methods receive the best performance among all strategies. The other finding is that the OVA method performs poorly in the present experiments for the considered classifiers, especially when the number of classes increases. In addition, the overall accuracy of all methods using a fixed classification algorithm generally decreases as the number of classes increases. This finding is expected as the difficulty of the multi-class problem increases when there are a large number of classes.

IV. Conclusions

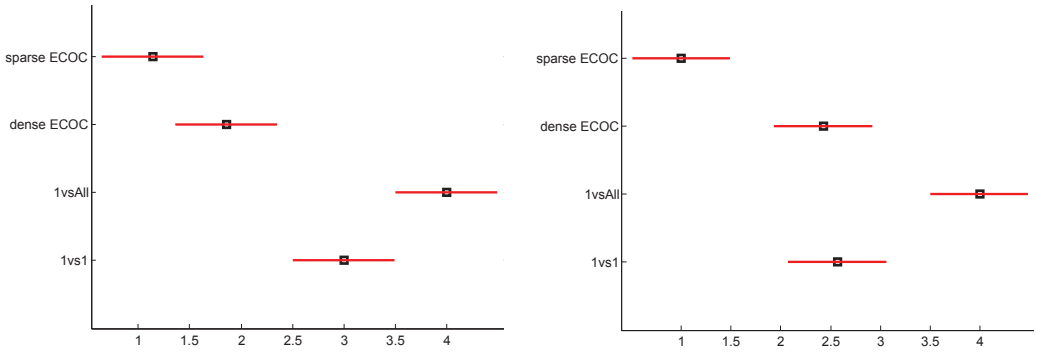
In this paper, we investigated the behavior of Error Correcting Output Codes on two image vision problems: logo recognition and shape classification using CART decision tree and AdaBoost as the base learners. These results indicate that even though the ECOC approach has received much less attention in the applied literature than classical OVO and OVA, it can significantly improve classification accuracy.

An analysis of the results shows a somewhat clearer picture. Using Decision Tree as the base learner, we found significant differences between ECOC and classical methods for both dense and sparse schemes. The ECOC methods are generally able to outperform OVO and OVA. Comparing the two schemes of ECOC, we can see that both achieve a similar performance, whereas dense ECOC is slightly inferior to sparse ECOC on the current datasets. Using AdaBoost as the



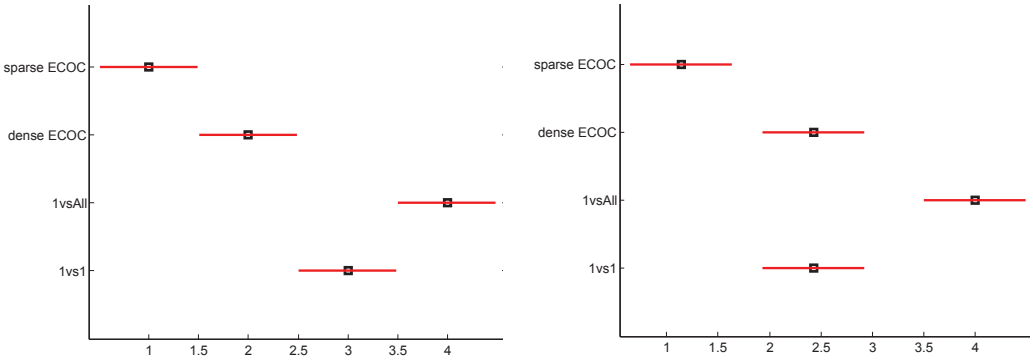
(a) CART Classifier (b) AdaBoost Classifier

Fig. 5. Average Accuracy of different methods using Logo dataset



(a) CART Classifier (b) AdaBoost Classifier

Fig. 6. Comparison results of rival methods based on the Nemenyi test using MPEG7 dataset



(a) CART Classifier (b) AdaBoost Classifier

Fig. 7. Comparison results of rival methods based on the Nemenyi test using Logo dataset

base learner, we can see that sparse ECOC achieved the best accuracy results. One can also see that the relative performance of OVO, dense ECOC, and sparse ECOC tends to be closer when the number of classes increases, which is consistent with findings from [1]. However, note that for a large number of classes the sub-linear number of classifiers of Dense and Sparse ECOC designs is considerably lower than OVO and OVA approaches.

REFERENCES

[1] N. Garcia-Pedrajas and D. Ortiz-Boyer, "An empirical study of binary classifier fusion methods for multiclass classification," *Information Fusion*, vol. 12, no. 2, pp. 111–130, 2011.

[2] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, ser. NIPS '97. Cambridge, MA, USA: MIT Press, 1998, pp. 507–513.

[3] P. Clark and R. Boswell, "Rule induction with cn2: Some recent improvements," in *Machine Learning EWSL-91*, ser. Lecture Notes in Computer Science, Y. Kodratoff, Ed. Springer Berlin / Heidelberg, 1991, vol. 482, pp. 151–163.

[4] R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *Neural Networks, IEEE Transactions on*, vol. 6, no. 1, pp. 117–124, 1995.

[5] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, p. 263286, 1995.

[6] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 285–297, 2009.

[7] E. Kong and T. Dietterich, "Error-correcting output coding corrects bias and variance," in *Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning*, A. Prieditis and J. Lemmer, Eds., 1995, pp. 313–321.

[8] ——, "Why error-correcting output coding works with decision trees," Technical Report, Department of Computer Science, Oregon State University, Corvallis, OR., Tech. Rep., 1995.

[9] T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multiclass learning problems," *Information Fusion*, vol. 4, no. 1, pp. 11–21, 2003.

[10] H. Chih-Wei and L. Chih-Jen, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.

[11] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: a stepwise procedure for building and training a neural network," in *Neurocomputing: Algorithms, Architectures and Applications*, J. Fogelman, Ed. Springer-Verlag, 1990.

[12] S.-H. Park and J. Fürnkranz, "Efficient pairwise classification," in *Proceedings of the 18th European Conference on Machine Learning (ECML 2007, Warsaw, Poland)*, J. N. Kok, J. Koronacki, R. López de Mántaras, S. Matwin, D. Mladenić, and A. Skowron, Eds. Springer-Verlag, 2007, pp. 658–665.

[13] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multiclass classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.

[14] M. Bagheri, Q. Gao, and S. Escalera, "Efficient pairwise classification using local cross off strategy," in *25th Canadian conf. on Artificial Intelligence*, Toronto, Canada, 2012, p. to appear.

[15] J. Ko and H. Byun, "Binary classifier fusion based on the basic decomposition methods," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, T. Windeatt and F. Roli, Eds. Springer Berlin / Heidelberg, 2003, vol. 2709, pp. 159–159.

[16] N. Garcia-Pedrajas and D. Ortiz-Boyer, "Improving multiclass pattern recognition by the combination of two strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 1001–1006, 2006.

[17] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2001.

[18] S. Escalera, A. Forns, O. Pujol, P. Radeva, G. Snchez, and J. Llads, "Blurred shape model for binary and grey-level symbol recognition," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1424–1433, 2009.

[19] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[20] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[21] R. Iman and J. Davenport, "Approximations of the critical regions of the friedman statistic," *Communications in Statistics*, vol. 6, pp. 571–595, 1980.