2014 22nd International Conference on Pattern Recognition

Generic Subclass Ensemble: A Novel Approach to Ensemble Classification

Mohammad ali Bagheri*† Qigang Gao *‡ Sergio Escalera§¶
* Faculty of Computer Science, Dalhousie University, Halifax, Canada
§ Centre de Visio per Computador, Campus UAB, Edifici O, Bellaterra, 08193 Barcelona, Spain
Email: †bagheri@cs.dal.ca, ‡ qggao@cs.dal.ca ¶sergio@maia.ub.es

***Abstract*—Multiple classifier systems, also known as classifier ensembles, have received great attention in recent years because of their improved classification accuracy in different applications. In this paper, we propose a new general approach to ensemble classification, named *generic subclass ensemble*, in which each base classifier is trained with data belonging to a subset of classes, and thus discriminates among a subset of target categories. The ensemble classifiers are then fused using a combination rule. The proposed approach differs from existing methods that manipulate the target attribute, since in our approach individual classification problems are not restricted to two-class problems. We perform a series of experiments to evaluate the efficiency of the generic subclass approach on a set of benchmark datasets. Experimental results with multilayer perceptrons show that the proposed approach presents a viable alternative to the most commonly used ensemble classification approaches.**

Keywords—Multiple classifier systems, ensemble classification, class decomposition, multiclass classification.

I. Introduction

The efficiency of pattern classification by a single classifier has been recently challenged by multiple classifier systems [1]–[5]. A multiple classifier system is a classification system made up of an ensemble of individual classifiers whose outputs are combined in some way to obtain a final classification decision. By combining a set of base classifiers, the combined efficiency of the ensemble of classifiers can compensate for a deficiency in one classifier. However, the ensemble approach depends on the assumption that single classifiers' errors are uncorrelated, which is known as classifier *diversity* in the background literature [6]. The intuition is that if each classifier makes different errors, then the total errors can be reduced by an appropriate combination of these classifiers.

The design process of a multiple classifier system generally involves two main stages: the collection of an ensemble of classifiers and the design of the combination rule [2]. Kuncheva summarized three primary approaches to creating an ensemble of classifiers [1]. These approaches can be considered as different ways to achieve diversity across base classifiers. The straightforward approach is to use different learning algorithms or variations of the parameters of the base learners. The second approach, which has been getting more attention in the ensemble literature, is to use different training sets to train base classifiers. Such sets are often obtained from the original training set by resampling techniques. The third approach is to train the individual classifiers with datasets that consist of different feature subsets [7], [8].

Another effective approach, which has not been paid much attention in the ensemble literature, is to "manipulate the target attribute", in which individual classifiers are built with different and usually simpler representations of the target classes [3], [9]. This approach was initially developed to solve the dilemma of extending binary classification algorithms to multiclass problems [10] and usually referred to as *"class binarization"* [11] in the multiclass classification literature. Existing methods based on this approach decompose the original multiclass problem into a series of smaller two-class problems. In this way, two-class problems can be solved by binary classifiers and their results can then be combined so as to provide a solution to the original problem. The procedure to decompose the multiclass problem into a set of binary problems is usually defined within the framework of Error Correcting Output Codes [12]–[15].

In this paper, we propose a new general approach to ensemble classification, named *generic subclass ensemble*, in which each base classifier is trained with data belonging to a subset of classes, and thus discriminates among a subset of target categories. The ensemble classifiers are then fused using a combination rule. The proposed approach differs from existing methods that manipulate the target attribute, since in our approach individual classification problems are not restricted to two-class problems. In light of this approach, class binarization techniques are considered special cases of the generic subclass ensemble approach. We also perform a series of experiments to evaluate the efficiency of the subclass approach on a set of benchmark datasets.

II. Generic subclass ensemble classification

In this section, we first introduce the generic subclass approach and explain its training and testing phases. Then, we propose three methods based on this approach.

A. The generic subclass approach

1) Training phase: In the training phase, different sub-problems are generated and a base classifier is trained on each sub-problem using samples belonging to a subset of the original set of classes. For each sub-problem, the subset of classes is divided into two or more meta-classes, where each meta-class consists of some combinations of the original classes. Accordingly, each classifier discriminates among classes that have been seen in its training. Here, similar to the ECOC framework, the process of decomposing the multiclass problem into a set of smaller binary/multiclass problems is represented by a matrix. This matrix is interpreted as a set of L learning

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.225

TABLE I. AN EXAMPLE OF A CLASS DECOMPOSITION MATRIX IN THE GENERIC SUBCLASS APPROACH.

	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}
ω_1	1	0	1	0	0	1	0	0	1	0
ω_2	1	0	0	1	0	2	1	2	1	0
ω_3	0	1	2	0	0	3	0	1	2	1
ω_4	2	2	3	0	1	0	2	0	1	1
ω_5	0	3	4	2	2	0	0	1	4	0
ω_6	0	3	2	0	1	0	2	1	5	2

problems, one for each column. Table I shows an example of a class decomposition matrix for a six-class problem that uses 10 classifiers. In this matrix, each column is associated with a subclass classifier, h_j, and each row is a unique codeword that is associated with an individual target class. For example, the second classifier, h_2, discriminates among samples of four classes: $\omega_3, \omega_4, \omega_5$ and ω_6. These classes are split into three meta-classes: $\{\omega_3\}, \{\omega_4\}$, and $\{\omega_5, \omega_6\}$. Similar to the sparse ECOC matrix [16], the zero value means that a given class is not considered in the training phase of a particular classifier

2) Testing phase: When testing an unlabeled pattern, x^*, each classifier casts a vote for one of the classes used in its training, creating an L long output code vector. This output vector is compared to each class codeword in the matrix, and the class whose codeword has the closest distance to the output vector is chosen as the predicted class. Similar to the ECOC method, the process of merging the outputs of individual classifiers is called decoding [17], [18]. Here, we propose a simple decoding technique:

$$y = \arg\min \frac{\sum_{i=1}^{L} w_i * sign(M(r,i)) * (M(r,i) <> h_i(x))}{sum(M(r,i) <> 0)},$$

where $sign(z)$ is +1 if $z > 0$, -1 if $z < 0$ and 0 otherwise and $a <> b$ is 1 if $a \neq b$ and 0 otherwise. Here, w_i represents each classifier' weight, which is set to the accuracy of classifier on the training data. $M(r,.)$ designates the codeword r in the matrix and $y \in \{1, \ldots, N_c\}$ is the predicted class.

B. Three methods based on the generic subclass approach

Utilizing the subclass approach will pose an important challenge: how to decompose the original multiclass problem into smaller problems? For a given problem of c classes, the number of valid partitions is $B_n - 1$, where B_n is the total number of partitions of a set with n members, named the Bell number.

Here, we propose three decomposition techniques: a problem-independent technique based on the exhaustive decomposition and two problem-dependent techniques based on the partitioning of the class space.

1) Exhaustive decomposition: One straightforward technique for class decomposition is to consider all possible partitions of classes, except the one that puts all classes in one partition. For problems with a large number of classes, however, the number of partitions and the number of selected classes for each sub-problem become very large. In order to limit the computational complexity, we employ two strategies to reduce the length of the codematrix, i.e. the number of classifiers. The first strategy is to have only one class in each meta-class. Second, we limit the number of classes to be taken for each classifier. Based on preliminary sets of experiments, choosing from two up to four classes for each sub-problem leads to high classification results. Still, for problems with a large number of classes, the number of possible permutations of three or four classes out of the number of classes increases dramatically. As a result, a large number of classifiers is required in the training and testing phases. In these cases, we propose a new version of exhaustive coding, named equidistant coding, where the distances between all pairs of class codewords are equal [1]. In this version, a subset of all permutations is chosen such that the sum of the number of each pair of classes is identical.

2) Decomposition based on a hierarchical clustering of class space: In this strategy, the class decomposition process is performed based on clustering of the class space that maximizes class discrimination in the patterns. Using a clustering technique, classes within each cluster are generally more similar which make them more difficult to be discriminated by a classifier. In this work, generating class subsets are guided through the hierarchical clustering of classes. The hierarchical clustering algorithm groups classes by creating a cluster tree or dendrogram. The dendrogram is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are merged in clusters at the next level. Each node of the cluster tree defines a partition of the classes. The partition at each node should be highly separable in terms of discrimination.

Using the generated cluster tree, the procedure of class partitioning, i.e. generating the class decomposition matrix, will be performed and the decomposition matrix will be generated. The procedure is composed of two main modules. Each module creates one decomposition matrix, and the final matrix is made by concatenating these two matrices.

In the first module, class partitioning is performed at different clustering levels ($1 < level < 6$). At each level, all classes are grouped into different partitions (clusters). An example of a clustering tree for a six class problem as well as the corresponding decomposition matrix is shown in Fig.1a. For instance, consider the horizontal line at the fifth level. This line intersects five lines of the dendrogram. These five lines partition the classes into five clusters: $\Psi_5^1 = \{c_1, c_3\}, \Psi_5^2 = \{c_2\}, \Psi_5^3 = \{c_4\}, \Psi_5^4 = \{c_5\}$, and $\Psi_5^5 = \{c_6\}$. The second module partitions classes under each internal node into two clusters. As an example, Fig.1b shows four internal nodes for a six-class problem. For example, the first node partitions classes below the right-hand side, namely c_5 and c_6, belonging to one cluster, while the class below the left-hand line, namely c_4, belongs to the other cluster. This partitioning is represented in the first column of the matrix.

3) Two methods based on hierarchical class partitioning: The clustering procedure involves defining a dissimilarity measure between objects in order to optimize the within- and between-cluster structure. Here, we employed two measures that aim to satisfy the condition of the high separability of classes: 1) the distance between the centroid of classes; and 2) the mutual information. Based on these two measures, two different methods are proposed. In the first method, the centroid of each class is computed as the average of patterns belonging to the corresponding class. The distance between

[1] distance is defined as the number of corresponding bits that differ.

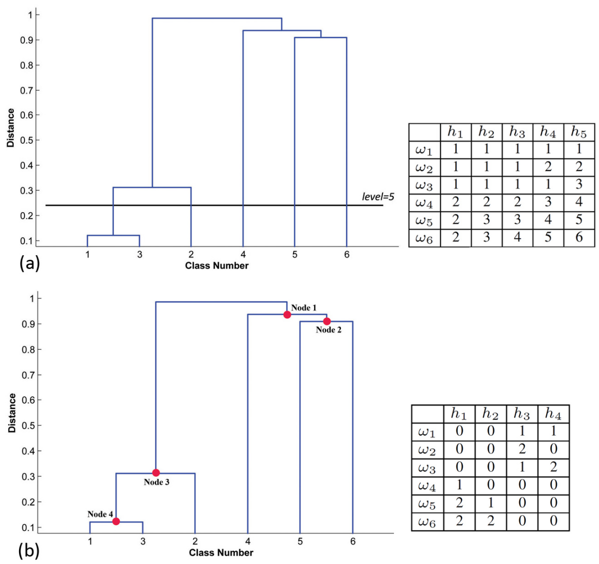

Fig. 1. An example of class partitioning using the hierarchical clustering technique; (a) class partitioning at different clustering level; (b). Class partitioning under each internal node.

two class centroids is a rough estimation of how separate two classes are. These centroid patterns are then hierarchically clustered. In the second method, the class portioning is based on the mutual information, which has been shown to be an effective criterion in terms of class separation. For more details on computing the mutual information between classes, please refer to [19].

III. Experimental comparison

A. Experimental settings

Data: The proposed generic subclass approach is evaluated on 15 multiclass datasets from the UCI machine learning repository [20]. Table II shows the number of classes, instances, and features of each dataset.

TABLE II. Summary of the datasets used.

Dataset	# Samples	# Classes	# Features
Abalone	4177	3	8
Cleafs	4758	8	64
Cmc	1473	3	9
Derm	358	6	34
Ecoli	336	8	7
Glass	214	6	9
Mfeat-fou	2000	10	76
Mfeat-pix	2000	10	240
Pendigits	10992	10	16
Sat	6435	6	36
Semeion	1593	10	256
Vowel	528	11	10
Waveforms	5000	3	40
Wine	178	3	13
Yeast	1484	10	8

Methods: We compare our proposed method with well-know ensemble classification methods, including bagging, AdaBoost, and RSM. For AdaBoost, we implemented the AdaBoost.M1 algorithm [21] which is a stable version of boosting for multiclass classification problems. The ensemble size, i.e. the number of base classifiers of the bagging and RSM and the number of iterations of the AdaBoost algorithm, is set to 25 [22].

In this study, a multilayer perceptron (MLP) with 10 hidden nodes and the hyperbolic tangent transfer function is chosen as the base learner.

Evaluation measurements: The classification accuracy is obtained by means of stratified 10-fold cross-validation.

TABLE III. Classification accuracies of the subclass-based methods

Datasets	EquiDist	ClsPart_Dist	ClsPart_MI
Abalone	66.41	65.62	65.31
Cleafs	75.08	78.53	79.18
Cmc	53.56	50.58	50.92
Derm	97.55	95.37	94.92
Ecoli	86.56	84.67	85.28
Glass	68.08	64.66	65.05
Mfeat-fou	84.25	79.65	80.20
Mfeat-pix	96.45	92.05	92.35
Pendigits	99.38	96.85	96.78
Sat	88.90	87.32	87.10
Semeion	92.48	79.75	81.32
Vowel	97.88	84.59	85.07
Waveforms	84.72	85.44	85.32
Wine	96.53	95.19	95.72
Yeast	60.40	56.90	58.05
Average	**83.22**	**79.81**	**80.17**

B. Determination of the best proposed method based on the generic subclass approach

As stated earlier, we proposed three different methods based on the subclass approach. The one based on the equidistant coding, named *Subclass.Equidistant*, and two methods based on the partitioning of the class space using two measures, the distance between class centroids and the mutual information, which we respectively named as *Subclass.ClsPart_Dist* and *Subclass.ClsPart_MI*.

Our first set of experiments is aimed at comparing these three methods. Table III shows the classification accuracies of these methods using the neural network as the base learner and Table IV shows the statistical comparison of these results based on the Wilcoxon sign rank test. These record present the number of datasets in which the algorithm in the row was better than the algorithm in the column (win), was worse (loss), or was equal (tie) As can be seen in these tables, Subclass.Equidistant performs better than the other two methods. The difference between Subclass.ClsPart_MI and Subclass.ClsPart_Dist is not significant, although the win/loss/tie record favors Subclass.ClsPart_MI.

TABLE IV. Statistical comparison of the results of the generic subclass-based methods using the Wilcoxon sign rank test.

	Subclass(ClsPart_dist)	Subclass (ClsPart_MI)
Subclass (Equidistant)	11/2/2	8/3/4
Subclass (ClsPart_dist)		1/3/11

In general, comparing the three versions of the generic subclass-based methods shows that in terms of classification accuracy the subclass ensemble based on the equidistance decomposition achieves better overall results. The main reason behind this improvement might be that by using the equidistant

TABLE V. CLASSIFICATION ACCURACIES OF DIFFERENT ENSEMBLE METHODS USING MLP NEURAL NETWORK AS THE BASE LEARNER.

Datasets	Single Classifier	Bagging	RSM	AdaBoost	Subclass (EquiDist)
Abalone	66.1	**67.25**	65.57	66.12	66.41
Cleafs	75.03	75.46	**78.13**	69.45	75.08
Cmc	52.34	**55.46**	52.20	53.02	53.56
Derm	94.53	97.16	97.37	97.43	**97.55**
Ecoli	84.72	**87.21**	84.4	81.76	86.56
Glass	62.94	67.75	**68.16**	67.84	68.08
Mfeat-fou	76.80	81.75	81.05	81.9	**84.25**
Mfeat-pix	81.75	92.8	95.4	92.55	**96.45**
Pendigits	92.11	96.19	91.31	99.02	**99.38**
Sat	86.70	87.33	87.75	88.31	**88.90**
Semeion	65.08	85.96	90.91	85.14	**92.48**
Vowel	65.62	81.70	75.92	94.36	**97.88**
Waveforms	85.34	85.92	**86**	84.5	84.72
Wine	94.86	97.76	**98.21**	97.32	96.53
Yeast	56.23	58.65	54.68	59.66	**60.4**
Average	**76.01**	**81.22**	**80.47**	**81.22**	**83.22**

TABLE VI. STATISTICAL COMPARISON OF THE DIFFERENT ENSEMBLE METHODS USING THE WILCOXON SIGN RANK TEST.

	Bagging	RSM	AdaBoost	Subclass(EquiDist)
Single classifier	0/12/3	0/9/6	3/10/2	0/10/5
Bagging		6/3/6	5/5/5	3/9/3
RSM			5/5/5	2/10/3
AdaBoost				2/9/4

coding more classifiers are usually generated. In ensemble systems, larger numbers of classifiers, especially when non-deterministic classifiers like neural networks are used as the base learner, usually lead to better classification accuracy. Similarly, the results of ECOC studies show that the ECOC method with longer codes is able to significantly improve the accuracy [11], [16]. In addition, in the second and third versions of the subclass approach, the class partitioning is performed by the hierarchical partition of classes that maximizes a discriminative criterion, i.e. the distance between class centroids in the second version and mutual information in the third version. However, there is no guarantee that this partitioning fits the underlying distribution of data. Therefore, the errors from a classifier at higher level are propagated to the lower levels. Due to the outperformance of the equidistant version of the subclass approach, in the following, we compare other ensemble methods with this version.

C. Comparison with standard ensemble methods

The average accuracy of the standard ensemble methods over 10 runs for each dataset is presented in Table V. For reference, we also show the accuracy of a single MLP classifier.

1) Statistical analysis of the classification results: Table VI shows the comparison of the different method using the Wilcoxon sign rank test. In this table, we show the win-loss-tie (WLT) comparison record of the algorithm in the column against the algorithm in the row.

The results in Tables V and VI indicate that overall the generic subclass approach obtained better results with many datasets. As an additional analysis, the improved accuracies of the Subclass.Equidistant method in comparison with other methods are shown in Fig. 2. In this figure, the obtained results are presented in the order of the number of dataset classes. The results of datasets with a larger number of classes are shown in order from the left side of the figure. From this arrangement, it can be seen that the generic subclass ensemble works better when there is a larger number of classes. In these cases, instead of combining individual classifiers trained with different subsets of samples or features, the more efficient approach is to train classifiers on a subset of classes. In this way, the problem that each classifier is going to be applied to is relatively smaller and can be solved more efficiently. On the other hand, when there is a large number of features, the subspace approach seems to be a good choice.

2) Analyzing the effect of ensemble size: Here, we investigate the performance of rival ensemble methods with different ensemble sizes; $1 \leq Ens.Size \leq 50$. For the generic subclass approach, however, the initial number of required classifiers for a given problem is fixed; which is equivalent to the number of columns of the class decomposition matrix. Therefore, to evaluate the performance of the ensemble system all classifiers should cast a vote.

In the case of the Subclass.Equidistant method, for problems with a large number of classes, the number of classifiers may be very large. In these cases, as we mentioned earlier, we limit the number of considered classes for each classifier to two or three. That is, each classifier discriminates between samples of two or three classes. For this design, the class decomposition matrix begins with all permutations of two classes, like the one-versus-one technique, and then continues until the number of classifiers is less than a pre-defined number, fixed at 50 in our experiments. On the other hand, for problems with small numbers of classes, the initial ensemble size is relatively small. One strategy to increase the number of classifiers is to duplicate the decomposition matrix. In this way, the ensemble size is the multiple of the initial number of classifiers, i.e. the length of codewords of the original class decomposition matrix. Even though the same sub-problems will be produced, the larger ensemble system can benefit from the variation of non-deterministic classifiers like neural networks.

Fig. 3 shows the classification accuracy of rival methods as a function of the ensemble size for 8 representative datasets. From these results, some general findings are summarized below:

- These experiments indicate that, in general, ensemble methods follow a similar trend. That is, their classification performance first improves as the ensemble size increases and then plateaus after a demarcation point, e.g. a value around 15-25. This observation is consistent with the results of many studies, see [7], [23], [24] as few examples.
- The underperformance of the Subclass.Equidistant method in problems with a small number of classes is mainly due to the significantly smaller number of classifiers. However, by increasing the ensemble size by duplicating the class decomposition matrix, classification accuracy was improved for many datasets.
- In problems with a larger number of classes, bagging, boosting, and RSM ensemble methods cannot continue to further improve with larger ensemble sizes. In these cases, the subclass approach shows the best performance.

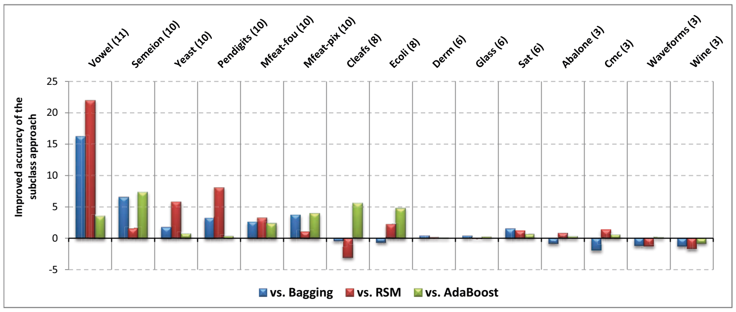

Fig. 2. The relative accuracy of different ensemble methods compared to the *Subclass.Equidistant* method.

IV. Conclusion

In this paper, we have proposed a new general approach to ensemble classification, named generic subclass ensemble. In this approach, an ensemble of classifiers is generated, in which each base classifier aims to discriminate between a subset of target categories. The proposed approach provides a general framework that can incorporate a wide range of class binarization techniques.

Based on the generic subclass approach, three methods are introduced: Subclass.Equidistant, Subclass.ClsPart_MI, and Subclass.ClsPart_Dist. Using the neural network as the base learner, we evaluated the efficiency of the generic subclass ensemble on a set of benchmark datasets. Experimental results show that the subclass approach presents a viable alternative to the most commonly used ensemble classification approaches. Specifically, this approach shows a better performance in problems with a larger number of classes. In these cases, instead of combining individual classifiers trained with different subsets of samples or features, the more efficient approach is to train classifiers on a subset of classes.

References

[1] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. New York, NY: Wiley, 2004.

[2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[3] L. Rokach, *Pattern classification using ensemble methods*, ser. Series in machine perception and artificial intelligence. Hackensack, N.J.; London: World Scientific, 2010.

[4] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman & Hall, 2012.

[5] M. A. Bagheri, G. Qigang, and S. Escalera, "A framework towards the unification of ensemble classification methods," in *12th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, 2013, pp. 351–355.

[6] T. Windeatt, "Accuracy/diversity and ensemble MLP classifier design," *TNN, volume = 17, number = 5, pages = 1194-1211, year = 2006*.

[7] T. K. Ho, "The random subspace method for constructing decision forests," *TPAMI*, vol. 20, pp. 832–844, 1998.

[8] D. W. Opitz, "Feature selection for ensembles," pp. 379–384, 1999.

[9] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4046–4072, 2009.

[10] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *TNNLS*, 2013.

[11] N. Garca-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.

[12] 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.

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

[14] M. Bagheri, Q. Gao, and S. Escalera, "Rough set subspace error-correcting output codes," in *IEEE International Conference on Data Mining (ICDM)*, Belgium, Brussels, 2012, pp. 822– 827.

[15] M. A. Bagheri, G. A. Montazer, and S. Escalera, "Error correcting output codes for multiclass classification: Application to two image vision problems," in *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, Shiraz, Iran, 2012, pp. 508–513.

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

[17] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *TPAMI*, vol. 32, no. 1, pp. 120–134, 2010.

[18] M. A. Bagheri, Q. Gao, and S. Escalera, "A genetic-based subspace analysis method for improving error-correcting output coding," *Pattern Recognition*, vol. 46, no. 10, pp. 2830–2839, 2013.

[19] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *JMLR*, vol. 3, pp. 1415–1438, 2003.

[20] C. Blake and C. Merz, "UCI repository of machine learning databases, department of information and computer sciences, university of california, irvine," 1998.

[21] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[22] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1, pp. 105–139, 1999.

[23] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *J. Artif. Res.*, vol. 11, pp. 169–198, 1999.

[24] D. Sun and D. Zhang, "Bagging constraint score for feature selection with pairwise constraints," *Pattern Recognition*, vol. 43, no. 6, pp. 2106–2118, 2010.

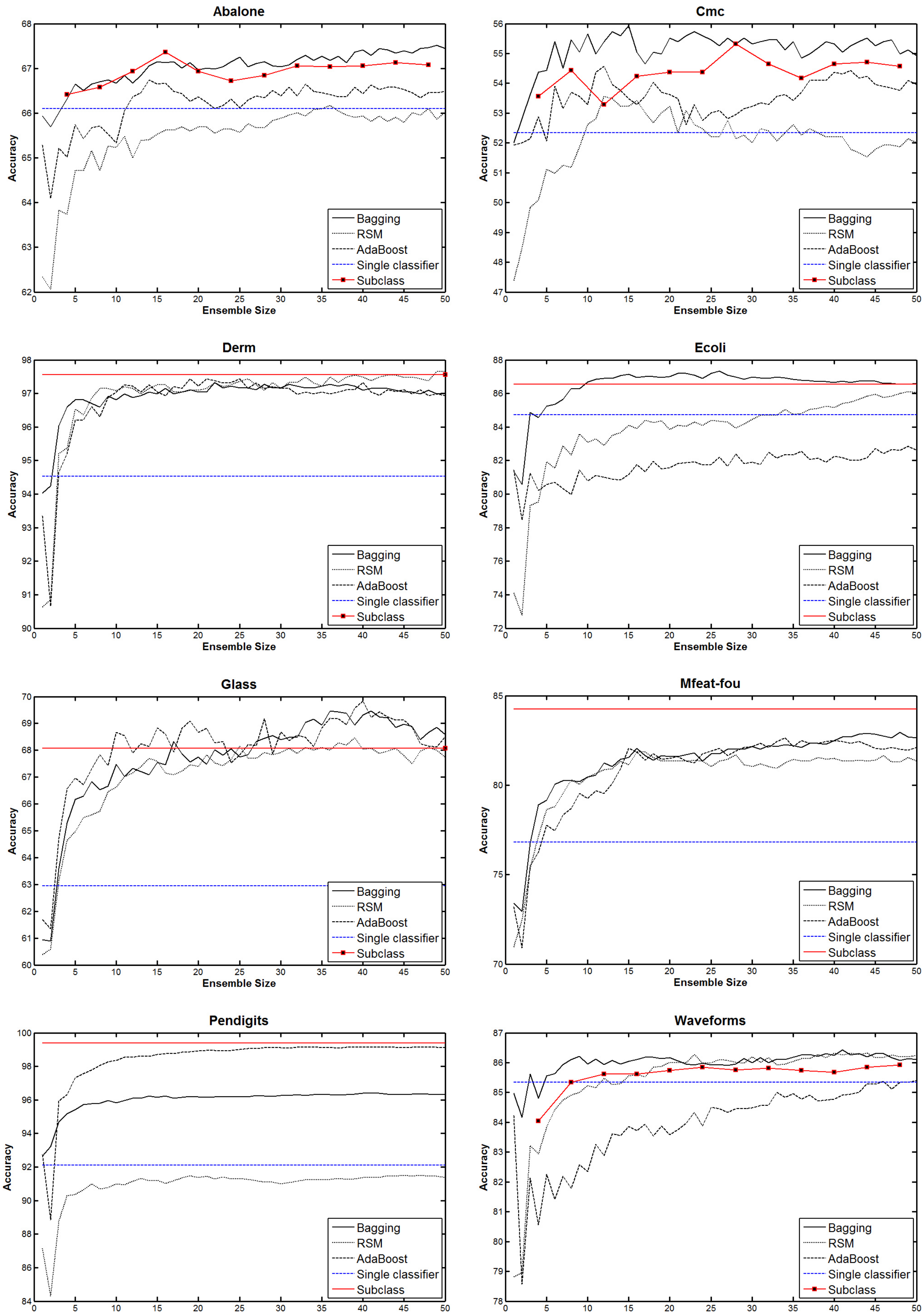

Fig. 3. Accuracy of ensemble classification methods versus the ensemble size.