A Diversity Production Approach in Ensemble of Base Classifiers

Hamid Parvin, Sara Ansari, and Sajad Parvin

Nourabad Mamasani Branch, Islamic Azad University, Nourabad Mamasani, Iran
{hamidparvin,s.ansari}@mamasaniiau.ac.ir

Abstract. One of crucial issue in the design of combinational classifier systems is to keep diversity in the results of classifiers to reach the appropriate final result. It's obvious that the more diverse the results of the classifiers, the more suitable final result. In this paper a new approach for generating diversity during creation of an ensemble together with a new combining classifier system is proposed. The main idea in this novel system is heuristic retraining of some base classifiers. At first, a basic classifier is run, after that, regards to the drawbacks of this classifier, other base classifiers are retrained heuristically. Each of these classifiers looks at the data with its own attitude. The main attempts in the retrained classifiers are to leverage the error-prone data. The retrained classifiers usually have different votes about the sample points which are close to boundaries and may be likely erroneous. Like all ensemble learning approaches, our ensemble meta-learner approach can be developed based on any base classifiers. The main contributions are to keep some advantages of these classifiers and resolve some of their drawbacks, and consequently to enhance the performance of classification. This study investigates how by focusing on some crucial data points the performance of any base classifier can be reinforced. The paper also proves that adding the number of all "difficult" data points just as boosting method does, does not always make a better training set. Experiments show significant improvements in terms of accuracies of consensus classification. The performance of the proposed algorithm outperforms some of the best methods in the literature. Finally, the authors according to experimental results claim that forcing crucial data points to the training set as well as eliminating them from the training set can lead to the more accurate results, conditionally.

Keywords: Classifier Fusion, Heuristic Retraining, Meta-Learner.

1 Introduction

The increasing importance of recognition systems arising in a wide range of advanced applications has led to the extensive study of classification [2], [4], [14] and [17]. So, a huge amount of research has been done around of it. While most of these researches have provided good performance for specific problem, they have not enough robustness for other problems. Due to the difficulty that these researches are faced to, recent researches are directed to the combinational methods that have more potential,

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part I, LNAI 7629, pp. 51–60, 2013.

robustness, resistance, accuracy and generality. Although the accuracy of the classifier ensemble is not always better than the most accurate classifier in ensemble pool, its accuracy is never less than average accuracy of them [9]. Combination of multiple classifiers (CMC) can be considered as a general solution for pattern recognition problems. Inputs of CMC are result of separate classifiers and output of CMC is their final combined decisions. Roli and Kittler [19] discuss that the rationale behind the Great tendency to multiple classifier systems (MCS) is due to some serious drawbacks of classical approach in designing of a pattern recognition system which focuses on the search for the best individual classifier. Identifying the best individual classifier for the classification task without deep prior knowledge is very difficult even impossible [3]. In addition, Roli and Giacinto [9] express that it is not possible to exploit the complementary discriminatory information that other classifiers may encapsulate with only a single classifier. It is worth-noting that the motivations in favor of MCS strongly resemble those of a "hybrid" intelligent system. The obvious reason for this is that MCS can be regarded as a special-purpose hybrid intelligent system.

In General, it is an ever-true sentence that "combining the diverse classifiers any of which performs better than a random classifier, results in a better classification". Diversity is always considered as a very important issue in classifier ensemble methodology. It is also considered as the most effective factor in succeeding an ensemble. The diversity in an ensemble refers to the amount of differences in the outputs of its components (classifiers) in deciding for a given sample. Assume an example dataset with two classes. Indeed the diversity concept for an ensemble of two classifiers refers to the probability that they may produce two dissimilar results for an arbitrary input sample. The diversity concept for an ensemble of three classifiers refers to the probability that one of them produces dissimilar result from the two others for an arbitrary input sample. It is worthy to mention that the diversity can converge to 0.5 and 0.66 in the ensembles of two and three classifiers respectively. Although reaching the more diverse ensemble of classifiers is generally handful, it is harmful in boundary limit. It means that increasing diversity in an ensemble to reach 0.5 results in an unsuccessful ensemble. It is a very important dilemma in classifier ensemble field: the ensemble of accurate/diverse classifiers can be the best. Although it's mentioned before the more diverse classifiers, the better ensemble, it is provided that the classifiers are better than a random classifier.

The authors believe that Combinational methods usually enhance the quality of the result of classification, because classifiers with different features and methodologies can eliminate drawbacks of each other. Kuncheva used Condorcet theorem to show that combination of classifiers could be able to operate better than single classifier. It was illustrated that if more diverse classifiers are employed in the ensemble, then error of them can considerably be reduced. Different categorizations of combinational classifier systems are presented in [10], [19]. Valentini and Masouli divide methods of combining classifiers into two categories: generative methods, non-generative methods. In generative methods, a set of base classifiers are created by a set of base algorithms or by manipulating dataset [9]. This is done in order to reinforce diversity of base classifiers. Generally, all methods which aggregate the primary results of the fixed independent classifiers are non-generative.

Neural network ensembles as an example of combinational methods in classifiers are also becoming a hot spot in machine learning and data mining recently [18]. Many researchers have shown that simply combining the output of many neural networks can generate more accurate predictions than that of any of those individual neural networks. Theoretical and empirical works show that a good ensemble is one in that the individual base classifiers have both accuracy and diversity, i.e. the individual base classifiers make their errors on different parts of the input space [6], [8].

2 Background

In generative approaches, diversity is usually made using two groups of methods. The first group obtains diverse individuals by training classifiers on different training set, such as bagging [1], boosting [9], [21], cross validation [8] and using artificial training examples [13].

Bagging [1] is the first and easiest resampling method. This meta-learner uses bootstrap sampling to produce a number of sub-samplings by randomly drawing, with replacement, N data points out of the train set (that has N data points). The individual classifiers are often aggregated by simple majority vote mechanism.

AdaBoost [25] successively produces a sequence of classifiers, where the training data points that are incorrectly classified by prior primary classifiers are chosen more frequently than data points which were properly classified. AdaBoost tries to create new classifiers that are capable of better forecasting of the data points which are misclassified by the existing ensemble. So it attempts to minimize the average miscalculation. Arc-X4 [26] belongs to the category of Boosting ensemble approaches.

The second group employs different structures, initial weighing, parameters and base classifiers to obtain various ensemble individuals. For example, Rosen [20] adapted the training algorithm of the network by introducing a penalty term to encourage individual networks to be decorrelated. Liu and Yao [12] used negative correlation learning to generate negatively correlated individual neural network.

There is another approach that is named selective approach where the diverse components are selected from a number of trained accurate base classifiers. For example, Opitz and Shavlik [16] have proposed a generic algorithm to search for a highly diverse set of accurate base classifiers. Lazarevic and Obradoric [11] have described a pruning algorithm to eliminate redundant classifiers. Navone et al. [15] have discussed another selective algorithm based on bias/variance decomposition. GASEN by Zhou et al. [22] and PSO based approach by Fu et al. [5] also has been introduced for selection of the ensemble components.

DECORATE is a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training examples. Comprehensive experiments have demonstrated that this technique is consistently more accurate than the base classifier, Bagging and Random Forests. Decorate also obtains higher accuracy than Boosting on small training sets, and achieves comparable performance on larger training sets [23]-[24]. In the rest of this paper, a new method to obtain diverse classifiers is demonstrated which uses manipulation of dataset structures.

Inspired from boosting method, in this paper a new sort of generative approaches is presented which creates new training sets from the original one. The base classifiers are trained focusing on the crucial and error prone data of the training set. This new approach which is called "Combination of Classifiers using Heuristic Retraining, CCHR" is described in section 2 in detail. In fact, the question of "how to create a number of diverse classifiers?" is answered in section 2. Section 3 addresses the empirical studies in which we show the great accuracy and robustness of CCHR method for different datasets. Finally, section 4 discusses the concluding remarks.

3 Proposed Method

The main idea of the proposed method is heuristically retraining of any base classifier on different subsets of training data. In CCHR meta-learner, the base classifiers are trained on some possible permutations of 3 datasets named: TS, NS, and EPS. They are abbreviation for Train Set, Neighbour Set and Error-Prone Set, respectively. The set involving TS and NS results in a classification by complex boundaries with more concentration on crucial points and neighbour of errors. The set involving TS, EPS and NS results in a classification by complex boundaries with more concentration on error prone (EPS) and crucial (NS) data points. The set involving TS and NS except the data points in EPS results in a classification by simple boundaries with more concentration on crucial points. The classifier trained on a set involving both TS and EPS leads to a classification by complex boundaries with more concentration on error prone data points. Finally the set involving TS except data points in EPS results in a classification by very simple boundaries.

Designing an ensemble of classifiers trained in these different defined sets, leads to an ensemble with a high degree of diversity. In the next step, the results of all these base classifiers are combined using simple average method.

At first, a base classifier is trained on TS. Then, using the trained base classifier, the data points that may be misclassified are recognized. This work is done for different perspectives of training-test datasets. It means that it is tried to detect all error-prone data on TS. It can be implemented using either leave-one-out technique or cross-validation technique.

In cross-validation which is also called the rotation method, an integer K (preferably a factor of N) is chosen and the dataset is randomly divided into K subsets of size N/K. Then, a classifier is trained on dataset-{*i*th subset of the dataset} and evaluated using *i*th subset. This procedure is repeated K times, choosing a different part for testing each time. When N=K, the method is called the leave-one-out or U-method.

In this paper, the dataset is decomposed into three partitions: training, evaluation and test subsets. The leave-one-out technique is applied to train set for obtaining the Error-Prone Set, EPS. As it is mentioned, using leave-one-out technique a base classifier on TS-{one of its data} is trained and evaluate whether the base classifier misclassifies that eliminated data point or not. If it is misclassified we insert it into EPS. It's obvious that, we repeat this work as many as the number of data points in training set. If training dataset is huge, the cross-validation technique can be used instead of leave-one-out technique, too.

In this work, the cross-validation technique is applied to {train set + validation set} for deriving the neighbor set, NS. Since the cross-validation is an iterative technique, for each iteration K-1 subset is considered as the training set and the remaining subset as validation set. The error on each validation set is added to an error set denoted by EPS. In the next step, for each member of the error set, the nearest neighbor data point which belongs to the same label of that member is found. We insert the nearest neighbor data point to each data point in error set into a neighbor set. This neighbor set is named NS.

The EPS and NS are obtained from previous section. In this section some base classifiers are trained based on them. The more diverse and accurate base classifiers, the better results in final. So, some combinations as shown in Table 1 are used to create diversity in our ensemble. The used permutations and the reasons of their usage are shown in Table 1. Training of base classifiers, using the combinations in Table 1, results in the classifiers that each of them focuses on a special aspect of data. This can result in very good diversity in the ensemble.

Table 1. Different data combinations and reasons of their usages

Num	TS	Resultant Classifier
1	TS	Creation of base classifiers
2	TS+NS	Classification by complex boundaries with more concentration on crucial points and neighbor of errors (NS)
3	TS+EPS+NS	Classification by complex boundaries with more concentration on error prone(EPS) and crucial points(NS)
4	TS-EPS+NS	Classification by simple boundaries with more concentration on crucial points
5	TS+EPS	Classification by complex boundaries with more concentration on error prone points (EPS)
6	TS-EPS	Classification by very simple boundaries

In this paper, 6 homogeneous base classifiers are trained with use of different data according to Table 1. Their outcomes are used in CCHR ensemble. CCHR algorithm is depicted in Fig 1.

```
NS: Neighbor Set, NS={};
EPS: Error Prone Set, EPS={};
Program CCHR
        1. NS=FindNS();            //calculating NS
        2. EPS=FindEPS();          // calculating EPS
        3. Train 6 base_classifiers according to Table 1.
        4. Combine the results using          simple average.
End.
```

Fig. 1. The proposed CCHR algorithm

After creating diverse classifiers for our classifier ensemble, the next step is finding a method to fuse their results and make final decision. Final decision is made in combiner part. Based on their output, there are many different combiner methods. Some traditional models of classifier fusion based on soft/fuzzy outputs are as follows:

1. Majority vote: assume that we have k classifiers. Classifier ensemble vote to class j if a little more than half of base classifiers vote to class j.
2. Simple average: the average results of separate classifiers are calculated and then the class with the most average value is selected as final decision.
3. Weighted average: it is similar to simple average except that a weight for each classifier is used to calculate the average.

In this paper, the simple average method is used to combine their results.

4 Experimental Results

The usual metric for evaluating an output of a classifier is accuracy; so the accuracy is taken as the evaluation metric throughout all the paper for reporting performance of classifiers. In all experimental results reported for CCHR meta-learner, Multi-Layer Perceptron (MLP) is taken as base classifier.

Table 2. Brief information about the used datasets

#	*Dataset Name*	*# of Class*	*# of Features*	*# of Samples*	*Data distribution per classes*
1	Iris	3	4	150	50-50-50
2	Halfrings	2	2	400	300-100
3	Wine	3	13	178	59-71-48

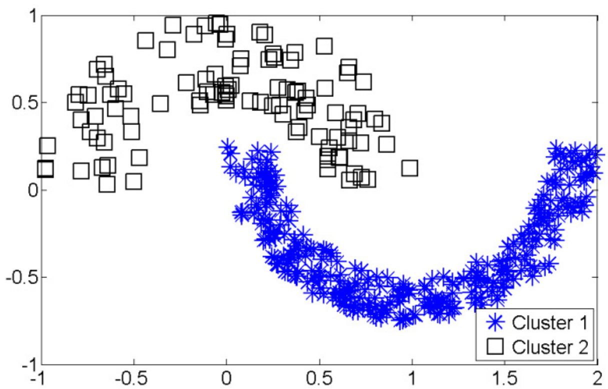

Fig. 2. Half Ring dataset

The proposed method is examined over 2 different standard datasets together with an artificial dataset. These real datasets are described in UCI repository [7]. Brief information about the used datasets is available in Table 2. The details of HalfRing dataset is available in [14]. The artificial HalfRing dataset is depicted in Fig. 2. The HalfRing dataset is considered as one of the most challenging dataset for the classification algorithms.

Table 3. Average results on Iris dataset

Train set	Classifier number as Table 1						CCHR
	1	2	3	4	5	6	
70%	95.01	95.20	95.20	94.97	**95.37**	95.07	**96.22**
50%	95.95	95.75	95.87	95.89	**96.24**	95.78	**96.64**
30%	93.57	93.26	93.17	93.64	**93.99**	93.48	**95.22**

This method is evaluated on two real standard datasets: Wine and Iris respectively in Table 3 and Table 4. All the presented results are reported over 10 independent runs. Result of each of classifiers is reported on 30%, 50%, 70% and 30%, 50% of Iris and Wine as training set, respectively. Table 3 and Table 4 show the base classifier that is trained on {TS+EPS} is relatively more robust than other base classifiers. This method is concentrated on error-prone data.

Table 4. Average results on Wine dataset

Train set	Classifier number as Table 1						CCHR
	1	2	3	4	5	6	
50%	91.58	91.64	92.66	91.98	**93.77**	91.29	**96.74**
30%	88.72	88.91	**89.31**	88.23	88.83	88.60	**93.76**

Table 5 shows the result of classification using CCHR meta-learner and two base classifiers comparatively. It is worthy to be mentioned that MLP is taken as the base classifier in CCHR meta-learner for reaching the results presented throughout all paper.

Table 5. CCHR vs. other methods

Classifier Type	Wine		Iris		
	50%	30%	70%	50%	30%
MLP	91.58	88.72	95.01	95.95	93.37
KNN	71.36	68.73	95.05	94.73	95.11
CCHR	**96.74**	**93.76**	**96.22**	**96.64**	**95.22**

As it is obvious from Table 5, recognition ratio is improved considerably by using CCHR meta-learner. Because of low number of features and records in Iris, the improvement is more significant on Wine dataset.

Table 6 shows the accuracy results of meta-learner CCHR and other traditional meta-learner methods comparatively. These results are reported averaging on the ten independent runs of the algorithms. In this comparison, the parameter K in K-Nearest Neighbor algorithm, KNN, is set to one. Also, the average accuracy of KNN method is reported over the 100 independent runs by randomly selecting a part of data as the training set, each time. To validate the meta-learner CCHR with harder benchmarks, an ensemble of the base MLPs is also implemented. These MLPs have the same structural parameters of the MLPs of meta-learner CCHR, i.e. two hidden layer with 10 and 5 neurons respectively in each of them. Like meta-learner CCHR, the voting method is chosen for combining their results.

Table 6. CCHR vs. other ensemble methods

	Wine			Iris			HalfRing		
	Train 30%	Train 50%	Train 70%	Train 30%	Train 50%	Train 70%	Train 30%	Train 50%	Train 70%
KNN	69.31	69.26	69.22	94.86	95.20	95.32	**100.00**	**100.00**	**100.00**
DT	84.80	90.26	92.34	92.03	93.47	95.23	94.49	97.42	99.41
MLP	88.72	91.58	93.09	93.37	95.95	95.01	94.27	95.26	95.05
Naïve Bayesian	**96.98**	96.42	**97.31**	95.01	95.51	95.57	94.11	94.85	94.53
Simple Ensemble	92.70	94.05	95.41	94.77	96.00	95.03	94.17	95.26	94.72
Random Forest	88.32	93.37	95.56	91.52	94.67	96.22	94.46	97.78	98.59
Decorate	96.31	95.43	95.47	93.88	95.82	**96.41**	95.57	95.71	97.27
AdaBoost	87.14	91.55	93.97	94.38	96.06	95.57	94.54	96.72	95.17
Arc-X4_1	96.4	96.13	96.42	94.86	96.07	95.33	97.08	98.02	98.32
Arc-X4_2	95.52	95.73	96.22	**95.33**	96.20	96.07	97.21	98.31	99.11
CCHR	93.76	**96.74**	96.56	95.22	**96.64**	96.22	99.19	**100.00**	**100.00**

The CCHR algorithm is compared with three state of the art meta-learners: decorate method, random forest method and boosting method. Here, the ensemble size of the random forest is 21. The ensemble size for Arc-X4_1 is 5 classifiers. While the ensemble size for Arc-X4_2 is 11 classifiers. The Simple Ensemble method is an ensemble of MLPs. The ensemble size in Decorate is 5. The classifiers used in Decorate are 5 MLPs. AdaBoost uses 11 MLPs with the same configuration of those used for CCHR meta-learner.

5 Conclusion and Future Work

In this paper, a new method for improving performance of multiple classifier systems named Combination of Classifiers using Heuristic Retraining, CCHR, is proposed. CCHR is based on heuristic retraining of base classifiers on different subsets of training data. Also, it is observed that different datasets result in different classifiers.

It is illustrated that the classifiers with complex boundaries and also those concentrating on error-prone data points have more generalization than others. It is shown that emphasizing on crucial data points causes improvement in results. With regard to the obtained results, we can conclude that the base classifier that is trained on {TS+EPS} is relatively more robust than other base classifiers. We can consider this base classifier as the best way for heuristically retraining of a base classifier. Also we have described that usage of different subsets of training set causes to a quite diverse ensemble.

Another interesting conclusion of the paper is the fact that emphasizing on the boundary data points, as boosting algorithm is not always very good. Although, boosting of the boundary data points in many cases is good, there are some problems where elimination of such data points is better. The Monk's problem is one of such cases which deleting error-prone data leads to a better result. Also, in data mining tasks which deal with huge data, the small size of ensemble is very interesting which is satisfied in CCHR as well. While CCHR only uses MLP as base classifier, it can use any other base classifier without losing its generality. So we can consider it as a meta-learner. For future work other base classifier can be used in CCHR meta-learner.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Parvin, H., Minaei-Bidgoli, B., Ghatei, S., Alinejad-Rokny, H.: An Innovative Combination of Particle Swarm Optimization, Learning Automaton and Great Deluge Algorithms for Dynamic Environments. International Journal of the Physical Sciences 6(22), 5121–5127 (2011)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, NY (2001)
4. Parvin, H., Minaei-Bidgoli, B., Alizadeh, H.: A New Clustering Algorithm with the Convergence Proof. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNCS, vol. 6881, pp. 21–31. Springer, Heidelberg (2011)
5. Fu, Q., Hu, S.X., Zhao, S.Y.: A PSO-based approach for neural network ensemble. Journal of Zhejiang University (Engineering Science) 38(12), 1596–1600 (2004) (in Chinese)
6. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Transaction on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
7. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998), `http://www.ics.uci.edu/mlearn/MLRepository.html`
8. Krogh, A., Vedelsdy, J.: Neural Network Ensembles Cross Validation, and Active Learning. Advances in Neural Information Processing Systems 7, 231–238 (1995)
9. Kuncheva, L.I.: Combining Pattern Classifiers, Methods and Algorithms. Wiley, New York (2005)
10. Lam, L.: Classifier Combinations: Implementations and Theoretical Issues. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 77–86. Springer, Heidelberg (2000)
11. Lazarevic, A., Obradovic, Z.: Effective pruning of neural network classifier ensembles. In: Proc. International Joint Conference on Neural Networks, vol. 2, pp. 796–801 (2001)
12. Liu, Y., Yao, X.: Evolutionary ensembles with negative correlation learning. IEEE Trans. Evolutionary Computation 4(4), 380–387 (2000)

13. Melville, P., Mooney, R.: Constructing Diverse Classifier Ensembles Using Artificial Training Examples. In: Proc. of the IJCAI 2003, pp. 505–510 (2003)
14. Minaei-Bidgoli, B., Parvin, H., Alinejad-Rokny, H., Alizadeh, H., Punch, W.F.: Effects of resampling method and adaptation on clustering ensemble efficacy (2011)
15. Navone, H.D., Verdes, P.F., Granitto, P.M., Ceccatto, H.A.: Selecting Diverse Members of Neural Network Ensembles. In: Proc. 16th Brazilian Symposium on Neural Networks, pp. 255–260 (2000)
16. Opitz, D., Shavlik, J.: Actively searching for an effective neural network ensemble. Connection Science 8(3-4), 337–353 (1996)
17. Parvin, H., Helmi, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., Shirgahi, H.: Linkage Learning Based on Differences in Local Optimums of Building Blocks with One Optima. International Journal of the Physical Sciences 6(14), 3419–3425 (2011)
18. Qiang, F., Shang-Xu, H., Sheng-Ying, Z.: Clustering-based selective neural network ensemble. Journal of Zhejiang University Science 6A(5), 387–392 (2005)
19. Roli, F., Kittler, J. (eds.): MCS 2002. LNCS, vol. 2364. Springer, Heidelberg (2002)
20. Rosen, B.E.: Ensemble learning using decorrelated neural network. Connection Science 8(3-4), 373–384 (1996)
21. Schapire, R.E.: The strength of weak learn ability. Machine Learning 5(2), 197–227 (1990)
22. Zhou, Z.H., Wu, J.X., Jiang, Y., Chen, S.F.: Genetic algorithm based selective neural network ensemble. In: Proc. 17th International Joint Conference on Artificial Intelligence, vol. 2, pp. 797–802 (2001)
23. Melville, P., Mooney, R.J.: Constructing Diverse Classifier Ensembles Using Artificial Training Examples. In: Conference on Artificial Intelligence, pp. 505–510 (2003)
24. Melville, P., Mooney, R.J.: Creating Diversity in Ensembles Using Artificial Data. Information Fusion: Special Issue on Diversity in Multiclassifier Systems 6(1), 99–111 (2004)
25. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory, pp. 23–37 (1995)
26. Breiman, L.: Arcing classifiers. The Annals of Statistics 26(3), 801–823 (1998)