Classifier Selection by Clustering

Hamid Parvin, Behrouz Minaei-Bidgoli, and Hamideh Shahpar

School of Computer Engineering, Iran University of Science and Technology (IUST),
Tehran, Iran
{parvin,b_minaei,shahpar}@iust.ac.ir

Abstract. This paper proposes an innovative combinational algorithm for improving the performance of classifier ensembles both in stabilities of their results and in their accuracies. The proposed method uses bagging and boosting as the generators of base classifiers. Base classifiers are kept fixed as decision trees during the creation of the ensemble. Then we partition the classifiers using a clustering algorithm. After that by selecting one classifier per each cluster, we produce the final ensemble. The weighted majority vote is taken as consensus function of the ensemble. We evaluate our framework on some real datasets of UCI repository and the results show effectiveness of the algorithm comparing with the original bagging and boosting algorithms.

Keywords: Decision Tree, Classifier Ensembles, Bagging, AdaBoosting.

1 Introduction

Although the more accurate classifier leads to a better performance, there is another approach to use many inaccurate classifiers specialized for a few data in the different problem spaces and using their consensus vote as the classifier. This can lead to a better performance due to the reinforcement of the consensus classifier in the error-prone feature spaces. In General, it is ever-true sentence that combining diverse classifiers usually results in a better classification [5].

This paper proposes a framework for development of combinational classifiers. In this new framework, a number of train data-bags are first bootstrapped from train data-set. Then a pool of weak base classifiers is created; each classifier is trained on one distinct data-bag. After that to get rid of similar base classifiers of the ensemble, using a clustering algorithm, here fuzzy k-means, the classifiers are partitioned. The partitioning is done considering the outputs of classifiers on train data-set as feature space. In each partition, one classifier, the head of cluster, is selected to participate in final ensmble. Then, to produce consensus vote, different votes (or outputs) are gathered out of ensmble. After that the weighted majority voting algorithm is applied over them. The weights are determined using the accuracies of the base classifiers on train dataset.

Decision Tree (DT) is one of the most versatile classifiers in the machine learning field. DT is considered as one of the unstable classifiers that can produce different

J.-F. Martínez-Trinidad et al. (Eds.): MCPR 2011, LNCS 6718, pp. 60–66, 2011.

results in its successive trainings on the same condition. It uses a tree-like graph or model of decisions. The kind of representation is appropriate for experts to understand what classifier does [8]. Its intrinsic instability can be employed as a source of diversity in classifier ensemble. The ensemble of a number of DTs is a well-known algorithm called Random Forest (RF) which is one of the most powerful ensemble algorithms. The algorithm of Random Forest was first developed by Breiman [2]. In this paper, DT is totally used as base classifier.

Rest of this paper is organized as follows. Section 2 is related works. In section 3, we explain the proposed method. Section 4 demonstrates results of our proposed method against traditional ones comparatively. Finally, we conclude in section 5.

2 Related Work

Generally, there are two important challenging approaches to combine a number of classifiers that use different train sets. They are Bagging and Boosting. Both of them are considered as two methods that are sources of diversity generation.

The term Bagging is first used by [2] abbreviating for Bootstrap AGGregatING. The idea of Bagging is simple and interesting: the ensemble is made of classifiers built on bootstrap copies of the train set. Using different train sets, the needed diversity for ensmble is obtained.

Breiman [3] proposes a variant of Bagging which it is called Random Forest. Random Forest is a general class of ensemble building methods using a decision tree as the base classifier. To be labeled a “Random Forest”, an ensemble of decision trees should be built by generating independent identically distributed random vectors and use each vector to grow a decision tree. In this paper Random Forest algorithm which is one of the well known versions of Bagging classifier [6] is implemented and compared with the proposed method.

Boosting is inspired by an online learning algorithm called Hedge(β) [4]. This algorithm allocates weights to a set of strategies used to predict the outcome of a certain event. At this point we shall relate Hedge(β) to the classifier combination problem. Boosting is defined in [4] as related to the “general problem of producing a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb.” The main boosting idea is to develop the classifier team D incrementally, adding one classifier at a time. The classifier that joins the ensemble at step k is trained on a dataset selectively sampled from the train dataset Z. The sampling distribution starts from uniform, and progresses towards increasing the likelihood of “difficult” data points. Thus the distribution is updated at each step, increasing the likelihood of the objects misclassified at step k-1. Here the correspondence with Hedge(β) is transposed. The classifiers in D are the trials or events, and the data points in Z are the strategies whose probability distribution we update at each step. The algorithm is called AdaBoost which comes from ADAptive BOOSTing. Another version of these algorithms is arc-x4 which performs as a newer version of ADAboost [6].

3 Proposed Framework

The main idea behind the proposed method is to use the most diverse set of classifiers obtained by Bagging or Boosting mechanism. Indeed a number of classifiers are first trained by the two well-known mechanisms: Bagging or Boosting. After that the produced classifiers partitioned according their outputs. Then a random classifier is selected from each of the produced clusters. Since each cluster is produced according to classifiers' outputs, it is highly likely that selecting one classifier from each cluster, and using them as an ensemble can produce a diverse ensemble that outperforms the traditional Bagging and Boosting, i.e. usage of all classifiers as an ensemble. Fig. 1 depicts the training phase of the Bagging method generally.

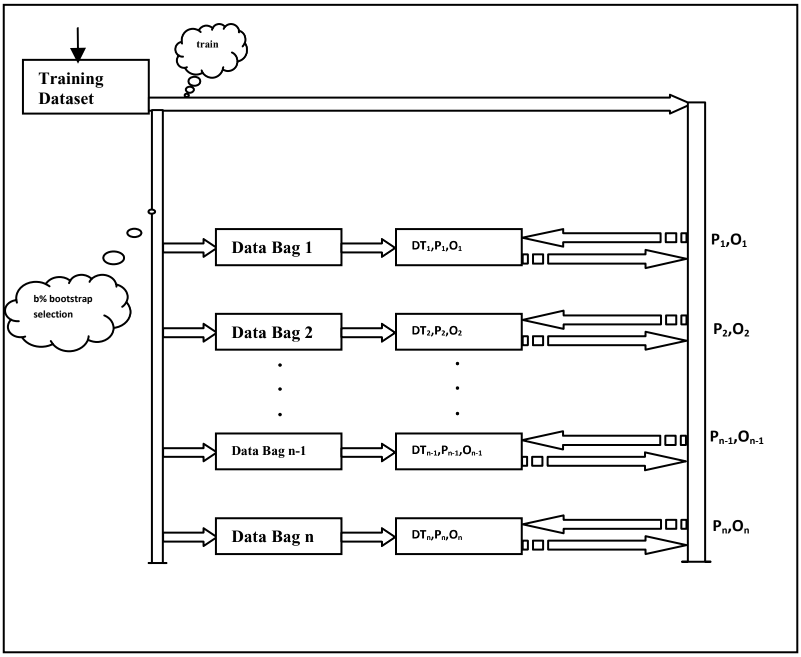

Fig. 1. Training phase of the Bagging method

As it is obvious from the Fig. 1, we bootstrap *n* subsets of dataset with *b* percent of the train dataset. Then a decision tree is trained on each of those subsets. We also then test each decision tree over the whole of train dataset and calculate its accuracy. The output of *i*th decision tree over train dataset is denoted by O_i and its accuracy is denoted by P_i. Fig. 2 depicts the training phase of the Boosting method. We again select a subset of dataset containing *b* percent of train dataset. Then the first decision tree is trained on this subset. After that the first classifier is tested on the whole train dataset which this results in producing the O_1 and P_1. Using O_1, the next subset of *b* percent of train dataset is obtained. This mechanism is continued in such a way that obtaining *i*th subset of *b* percent of train dataset is produced considering the O_1, O_2, …, O_{i-1}. For more information about the mechanism of Boosting, the reader can refer to Kuncheva [6].

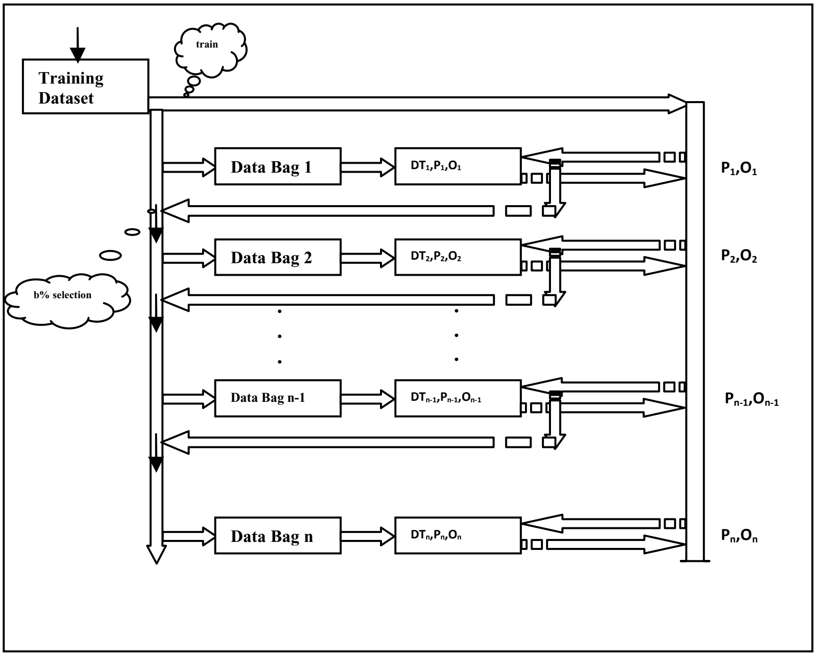

Fig. 2. Training phase of the Boosting method

The proposed method is generally illustrated in the Fig. 3. In the proposed method we first produce a dataset whose *i*th dataitem is O_i. Features of this dataset are real dataitems of under-leaning dataset. Then we have a new dataset having *n* classifiers and *N* features, where *n* is a predefined value showing the number of classifiers produced by Bagging or Boosting and *N* is the cardinality of under-leaning datasets. After producing the mentioned dataset, we partition that dataset by use of the clustering algorithm which results in some clusters of classifiers. Each of the classifiers of a cluster has similar outputs on the train dataset; it means these classifiers have low diversities, so it is better to use one of them in the final ensemble rather than all of them. For escaping from outlier classifiers, we ignore from the clusters which contain number of classifiers smaller than a threshold.

Let us assume that *E* is the ensemble of *n* classifiers $\{DT_1, DT_2, DT_3 \ldots DT_n\}$. Also assume that there are *m* classes in the case. Next, assume applying the ensemble over data sample *d* results in a binary *D* matrix like equation 1.

$$D = \begin{bmatrix} d_{1\ 1} & d_{1\ 2} & . & d_{1\ n} \\ . & . & . & . \\ d_{m-1\ 1} & d_{m-1\ 2} & . & d_{m-1\ n} \\ d_{m\ 1} & d_{m\ 2} & . & d_{m\ n} \end{bmatrix} \quad (1)$$

where $d_{i,j}$ is one if classifier j votes that data sample d belongs to class i. Otherwise it is equal to zero. Now the ensemble decides the data sample d to belong to class q according to equation 2.

$$q = \arg\max_{i=1}^{m} \left| \sum_{j=1}^{n} w_j * d_{i\ j} \right| \tag{2}$$

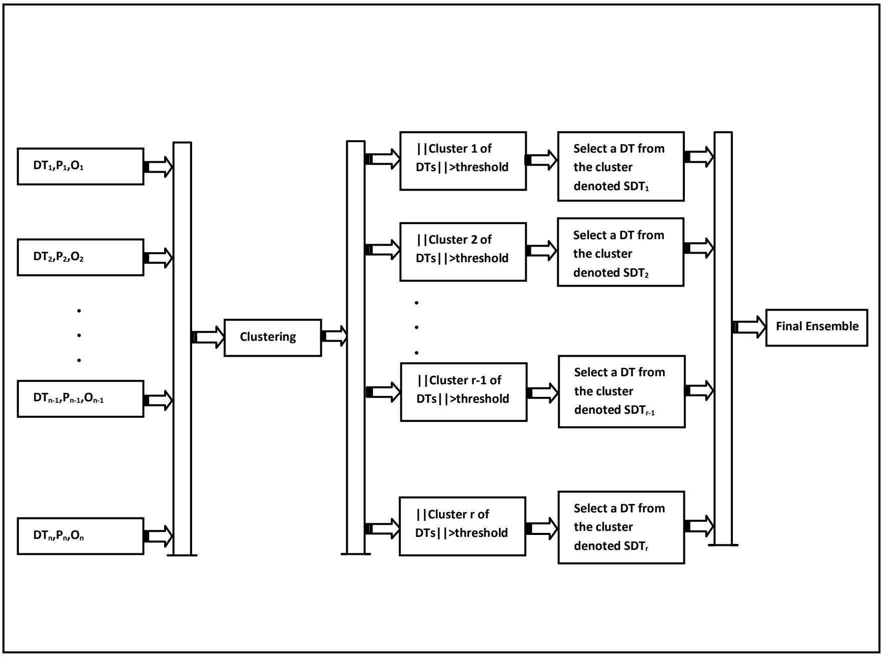
DT_1,P_1,O_1
DT_2,P_2,O_2
DT_{n-1},P_{n-1},O_{n-1}
DT_n,P_n,O_n
Clustering
||Cluster 1 of DTs||>threshold
||Cluster 2 of DTs||>threshold
||Cluster r-1 of DTs||>threshold
||Cluster r of DTs||>threshold
Select a DT from the cluster denoted SDT_1
Select a DT from the cluster denoted SDT_2
Select a DT from the cluster denoted SDT_{r-1}
Select a DT from the cluster denoted SDT_r
Final Ensemble

Fig. 3. Proposed method for selecting the final ensemble from a pool of classifier generated by Bagging or Boosting

where w_j is the weight of classifier j which is obtained optimally according to equation 3 [6].

$$w_j = \log \frac{p_j}{1 - p_j} \tag{3}$$

where p_j is accuracy of classifier j over total train set. Note that a tie breaks randomly in equation 2.

4 Experimental Results

This section evaluates the result of applying proposed algorithm on some real datasets available at USI repository [1] and one hand made dataset named half-ring. The details of half-ring dataset can be available in [7]. These dataset are summarized in the Table 1.

Table 1. Details of used dataset

Dataset Name	# of dataitems	# of features	# of classes
breast cancer	404	9	2
bupa	345	6	2
glass	214	9	6
galaxy	323	4	7
half-ring	400	2	2
heart	462	9	2
ionosphere	351	34	2
iris	150	4	3
test monk1	412	6	2
test monk2	412	6	2
test monk3	412	6	2
train monk1	124	6	2
train monk2	124	6	2
train monk3	122	6	2
wine	178	13	3

Measure of decision in each employed decision tree is taken as gini measure. The threshold of pruning is set to 2. Also the classifiers' parameters are fixed in all of their usages.

In all experiments *n*, *r*, *b* and threshold of accepting a cluster are set to 151, 11, 30 and 2 (i.e. only the clusters with one classifier is dropped down) respectively. All the experiments are done using 4-fold cross validation. Clustering is done by fuzzy kmeans with *r* (11) clusters. Table 2 shows the accuracies of different methods.

Table 2. Comparison of the results. * shows the dataset is normalized, and 4 fold cross validation is taken for performance evaluation. ** shows that the train and test sets are predifined.

	Arc-X4	Random Forest	Proposed Random Forest	Proposed Arc-X4
breast cancer*	96.18	95	**96.47**	95
bupa*	71.51	68.31	**72.97**	66.28
glass*	65.09	62.26	**66.23**	60.85
galaxy*	70.94	69.06	**72.5**	67.5
half-ring*	**97.25**	95.75	**97.25**	95.75
heart*	70.87	68.26	**72.61**	68.26
ionosphere*	**92.24**	90.52	91.54	90.52
iris*	**95.95**	**95.95**	**95.95**	95.27
monk problem1**	98.11	97.49	**98.76**	97.37
monk problem2**	97.01	86.64	**97.62**	86.73
monk problem3**	87.29	**96.92**	87.34	96.34
wine*	96.59	92.61	**98.3**	93.18
Average	86.59	84.9	**87.3**	84.42

While we choose only at most 7.3 percent of the base classifiers of Random Forest, the accuracy of their ensemble outperforms the full ensemble of them, i.e. Bagging. Also it outperforms Boosting.

Because the classifiers selected in this manner (by Bagging along with clustering), have different outputs, i.e. they are as diverse as possible, they are more suitable than their all ensemble. It is worthy to mention that the Boosting is inherently diverse enough to be an ensmble totally; and the reduction of ensmble size by clustering destructs their Boosting effect. Take it in the consideration that in Boosting ensmble, each member covers the drawbacks of the previous ones.

5 Conclusion and Future Works

In this paper, we have proposed a new method to improve the performance of classification. The proposed method uses Bagging as generator of the base classifiers. Then using fuzzy kmeans we partition the classifiers. After that we select one classifier per a validated cluster.

While we choose only at most 7.3 percent of the base classifiers of Bagging, the accuracy of their ensemble outperforms the full ensemble of them. Also it outperforms Boosting.

As a future work, we can turn to research on the variance of the method. Since it is said about Bagging can reduce variance and Boosting can simultaneously reduce variance and error rate.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Breiman, L.: Bagging Predictors. Journal of Machine Learning 24(2), 123–140 (1996)
3. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
4. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55(1), 119–139 (1997)
5. Gunter, S., Bunke, H.: Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. IWFHR (2002)
6. Kuncheva, L.I.: Combining Pattern Classifiers, Methods and Algorithms. Wiley, New York (2005)
7. Minaei-Bidgoli, B., Topchy, A.P., Punch, W.F.: Ensembles of Partitions via Data Resampling. In: ITCC, pp. 188–192 (2004)
8. Yang, T.: Computational Verb Decision Trees. International Journal of Computational Cognition, 34–46 (2006)