A Novel Adaptive-Boost-Based Strategy for Combining Classifiers Using Diversity Concept

Conference Paper · January 2007

DOI: 10.1109/ICIS.2007.37 · Source: DBLP

Abbas Golestani
University of Windsor

A Novel Adaptive-Boost-Based Strategy for Combining Classifiers Using Diversity Concept

Abbas Golestani [1,2] Kushan Ahmadian Ali Amiri [1,2] MohammadReza JahedMotlagh
abbas_golestani@comp.iust.ac.ir, ku_ahmadian@comp.iust.ac.ir, , jahedmr@iust.ac.ir
1- Islamic Azad University – Zanjan branch
2- Computer faculty of Iran University of Science and Technology

Abstract

In classifiers combination, the diversity rate among classifier's outputs is one of the most important discussions. There are different methods for combining classifiers. AdaBoost is an incremental method for creating a classifiers ensemble in which every AdaBoost algorithm has a local centrality. It means that classifiers are data biased and classify special data. In this paper we intend to find a new method for combining classifiers by using AdaBoost method and diversity concept. We have checked this method over different data sets and compared results of this method with others. These results indicate that we can develop other versions of this method for achieving a better performance.

1. Introduction

The optimal classifier in every case is highly dependent upon the problem domain. In practice, one might come across a case where no single classifier can achieve an acceptable level of accuracy. In such cases, it would be better to pool the results of different classifiers to achieve the optimal accuracy. Every classifier operates well on different aspects of the training or test feature vector. As a result, assuming appropriate conditions and combining multiple classifiers may improve classification performance when compared with any single classifier.

Instead of looking for the best set of features and the best classifier, now we look for the best set of classifiers and then the best combination method [1].

If we have a perfect classifier that makes no errors, then we do not need an ensemble. If, however, the classifier does make errors, then we seek to complement it with another classifier, which makes errors on different objects. The diversity of the classifier outputs is therefore a vital requirement for the success of the ensemble. Intuitively, we want the ensemble members to be as correct as possible, and in case they make errors, these errors should be on different objects.

In recent years, a new method has been proposed which covers shortcomings of the previous methods of combining classifiers. This new method is called AdaBoost [2]. In this method, choice of training data for the new classifier depends on the previous classification fault over those patterns. Using AdaBoost method and diversity concept between classifiers, we proposed a new method that increases accuracy and decreases error. We have called this method 2-Adaboost.

In part 2 we describe AdaBoost method for the sake of familiarity with the bases of this algorithm. In part 3, we discuss the diversity concept between classifiers. In part 4, we examine the new method (2-Adaboost) in details and in part 5 we compare the results of the new method with other methods.

2. Adaboost Method

AdaBoost Constructs classifiers by modifying the training set based on the previous classifier's performance. It does this by getting the new classifier to put more emphasis on those objects which the previous classifier found difficult to classify accurately [2]. This is achieved by maintaining a distribution of weights over the training set, which can be modified as required on each iteration. Some implementations of AdaBoost use a resampling method [3] and others use re-weighting [4]. These differ according to whether you resample from the original training set or attach weights to each data point and re-use the whole training set to build the next classifier. The choice of implementation does not affect AdaBoost too much although boosting with re-weighting is a more direct implementation of the theory [2, 4]. Research by

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

Breiman suggests that there is very little difference in the results obtained using the two methods [3]. For the implementation, each weight determines the probability of its associated pattern being selected for the training set for an individual component classifier [5]. Initially all weights are set equal. On each round if a training pattern is not accurately classified then its chances of being selected again for a subsequent training set are increased by increasing the value of its associated weight [2, 6]. In this way the next classifier is forced to concentrate on more difficult samples of the training set. Each AdaBoost algorithm has a local centrality. For example, for using a special subset of data set, classifiers in AdaBoost algorithm perform better on special data. This property has been shown in figure 1.

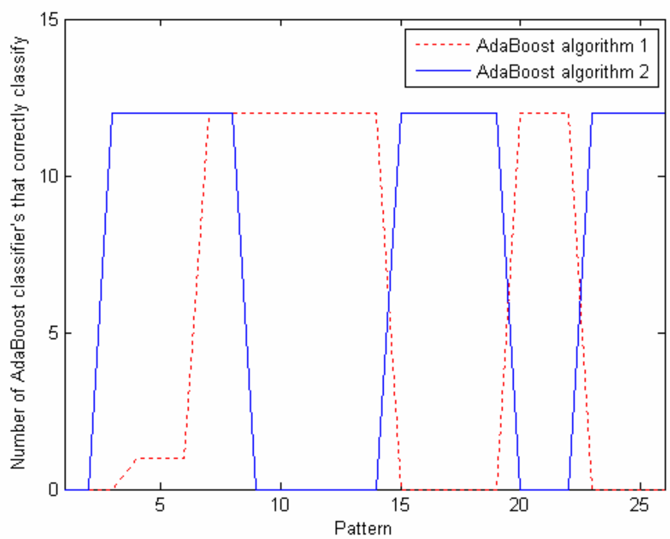

Figure 1. Local centrality for each AdaBoost algorithms.

3. Diversity measures

The diversity among the combination of classifiers is defined as: if one classifier has some errors, then for combination, we look for classifiers which have errors on different objects [7]. Among a set of classifiers, basically, there are two kinds of diversity measure methods, pairwise and nonepairwise.

3.1 Pairwise Measures

These measures, and the ones discussed by hitherto, consider a pair of classifiers at a time. An ensemble of L classifiers will produce $\frac{L(L-1)}{2}$ pair wise diversity Values. To get a single value we average across all pairs.

3.1.1 The Disagreement Measures:

The disagreement measure is probably the most intuitive measure of diversity between a pair of classifiers. Table1 shows the classification probability of two classifiers *I* and *k* for a given sample. The probability that the two classifiers will disagree on their decisions is considered as the diversity between two classifiers [8, 9]. Therefore, $D_{i,k} = (b+c)$.

Table 1. The confusion matrix of two classifier D_i , D_k

	D_k correct(1)	D_k wrong(0)
D_i correct(1)	a	b
D_i wrong(0)	c	d

3.2 Nonpairwise Measures

The measures of diversity introduced below consider all the classifiers together and calculate directly one diversity value for the ensemble.

3.2.1 Entropy evaluation method:

For a particular $z_j \in Z$ when $\lfloor L/2 \rfloor$of the votes are 0s (1s) and the other $L-\lfloor L/2 \rfloor$ votes are 1s (0s). If they all were 0s or all were 1s, there is no disagreement, and the classifiers cannot be deemed Diverse. One possible measure of diversity based on this concept is

$$E = \frac{1}{N}\ \frac{2}{L-1} \sum_{j=1}^{N} \min\{ (\sum_{i=1}^{L} y_{i,j}) , (L-\sum_{i=1}^{L} y_{i,j}) \}$$

E varies between 0 and 1, where 0 indicates no difference and 1 indicates the highest possible diversity. Let all classifiers have the same individual accuracy p. Then while value 0 is achievable for any number of classifiers L and any p, the value 1 can only be attained for $p \in \left[\frac{L-1}{2L}, \frac{L+1}{2L}\right]$ [9].

This method has some limitations, for example the precision of all classifiers must be *p*.

Generally, there are 10 famous methods for measuring diversity which are shown in Table 2. The symbol (↑) means, increasing the output of respective methods shows more diversity and the symbol (↓) means, decreasing the output of respective methods shows more diversity [10].

4. 2-AdaBoost method

This method increases accuracy of classifiers set when applied to a specific data set. In the original Adaptive boost method, classifiers are being trained based on the error made by the previous classifiers. However as our experiments showed, it leads to a premature concentration on different parts of data. It means that classifiers trained with adaboost sets have coincident error. So common intuition suggests that classifiers of ensemble should be as accurate as possible and should not make coincide errors. We reach this property if attend to diversity of ensemble.

Basically in our experiments we had used homogeneous classifiers (Knn) which resulted in less diverse ensembles. After a few experiments with some sets of classifiers we found that using different types of classifiers results in better performance for creating ensembles due to the diversity property of different types of classifiers.

In this method, if we have different types of classifiers (for example 5 classifiers Knn(k=1), Knn(k=3), svm(α=1), anorm, kde) , each classifier has been tested over specified data, so we calculate diversity between all dual combination of 5 classifiers. We consider the classifier pair that has more diversity and each classifier in this specific pair constructs kernel of an AdaBoost algorithm and by combining these two AdaBoost sets we obtain the most efficient combination. Figure 2 shows process of this new method.

Table 2. List of some popular diversity measures

Name	Notation	Type	Pairwise or Nonpairwise
Q-statistic[11]	Q	↓	*Pairwise*
Correlation coefficient[12]	ρ	↓	*Pairwise*
Disagreement[13,14]	D	↑	*Pairwise*
Double-fault[15]	DF	↓	*Pairwise*
Kohavi-Wolpert variance[16]	Kw	↑	*Nonpairwise*
Measurement of interrater agreement [17]	κ	↓	*Nonpairwise*
Entropy[18]	Ent	↑	*Nonpairwise*
Measure of difficulty[19]	θ	↓	*Nonpairwise*
Generalized diversity[20]	GD	↑	*Nonpairwise*
Coincident failure diversity[21]	CFD	↑	*Nonpairwise*

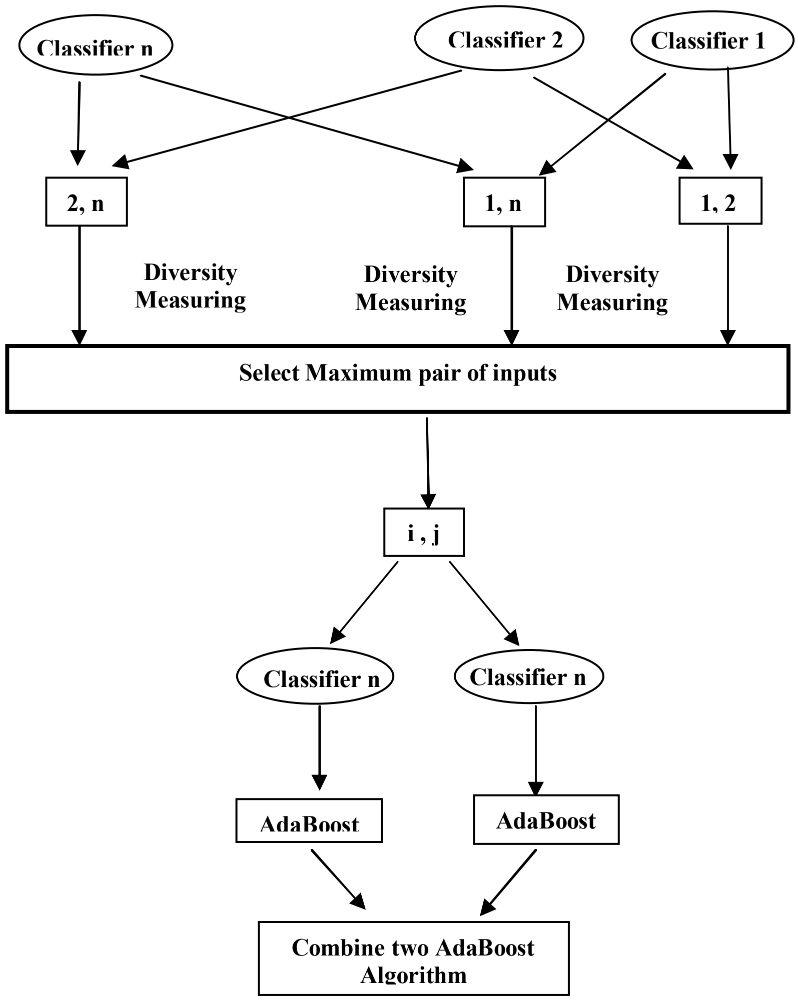

Figure 2. Process of New Algorithm (2-AdaBoost Algorithm)

5. Experimental result

Using 2-AdaBoost algorithm over specified data sets leads to results that have been listed bellow. The first data set is ***Pima*** data set that has 764 patterns with 8 features. Two classes exist in this data set. Next data set is ***Spam*** data set that has 4016 patterns with 54 features. Finally we test ***Haberman*** data set and ***Horse-colic*** data set. These data sets have 2 classes too. Five classifiers have been used as the base classifiers and then obtained results of this algorithm have been compared over these 2 data sets. Base classifiers are: *Knn (k=1), Knn (k=3), Svm (α =1), Anorm, Kde*.

5.1 Experimental results with *Pima* data set

First, each base classifier has been tested over this data set and the error rate of the classifiers has been measured. This experiment is shown in table 3.Then the AdaBoost algorithm has been implemented over each base classifier and has been tested over data. We have measured the error rate of each AdaBoost classifier. This result is shown in Table 4.

Now, we calculate diversity between all dual combinations of 5 classifiers, therefore we hold 2 classifiers that have more diversity and use our new algorithm over them. In this experiment *Kde* and *Anorm* classifiers have more diversity.

2-AdaBoost method uses *Kde* and *Anorm* classifiers as the kernel of the algorithm. So, the error rate of the new algorithm has been listed bellow:
Error rate: 11.68%

5.2 Experiment with *Spam* data set

Each base classifier has been tested over this data set similar previous experiment and error rate of classifiers has been measured. This experiment is shown in table 3. Similar previous experiment, AdaBoost algorithm has been implemented over each base classifier and has been tested over data. We have measured the error rate of each AdaBoost classifier. This result is shown in table 4.

Now, we calculate diversity between all dual combination of 5 classifiers, therefore we hold 2 classifiers that have more diversity and use the new algorithm over them. In this experiment *Kde* and *Anorm* classifiers have more diversity.

2-AdaBoost method uses *Kde* and *Anorm* classifiers as the kernel of algorithm. So, error rate of new algorithm has been listed bellow:

Error rate: 5.63%

5.3 Experiment with other data sets

Similar to previous experiment we have used Haberman and Horse-colic data sets. First we use the base classifiers for these data sets. After this we made Adaboost set by the base classifiers. In last we used 2-Adaboost algorithm. In these data sets SVM and Anorm classifiers had the highest diversity. The error rate of 2-Adaboost method has been listed bellow respectively.

Error rate of 2-Adaboost method over Haberman data set: 4.5%

Error rate of 2-Adaboost method over Horse-colic data set: 7.06%

Table 3. Error rate of base classifiers using different dataset

classifiers	Knn(k=1)	Knn(k=3)	Svm	Kde	Anorm
Error rate in Pima	33.77 %	36.36%	61.04%	38.96%	61.04%
Error rate in Spam	22.08%	28.57%	62.34%	10.39%	62.34%
Error rate in Haberman	21.89%	19.60%	26.14%	15.68%	17.97%
Error rate in HorseColic	35.32%	32.06%	27.98%	31.25%	38.85%

Table 4. Error rate of AdaBoost Algorithms using different data set

Classifiers / Error	AdaBoost with kernel Knn(k=1)	AdaBoost with kernel Knn(k=3)	AdaBoost with kernel Svm	AdaBoost with kernel Kde	AdaBoost with kernel Anorm	**2-Adaboost Method**
Pima	38.96%	49.35%	61.04%	33.77%	61.04%	**11.68%**
Spam	28.57%	42.86%	61.90%	29.44%	61.90%	**5.63%**
Haberman	19.60%	2156%	22.87%	13.72%	11.43%	**4.57%**
HorseColic	27.17%	29.89%	20.38%	28.53%	25.27%	**7.06%**

6. Conclusion

We proposed a new method for combining classifiers based on the AdaBoost algorithm and diversity measure between classifiers. AdaBoost algorithm and diversity explained in section 2 and 3. Experimental results with four data sets (Pima, Spam, Haberman and Horse-colic) from UCI ML repository has been discussed. These results show the better performance of the new method. We found that every AdaBoost algorithm concentrate on a special kernel of a given data set. Combining 2 independent AdaBoost

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

algorithms that have enough diversity could result better coverage of the patterns. In this paper we combined 2 independent AdaBoost algorithms, for future work it might be better that one combines more AdaBoost algorithms or find a relation between the number of combined AdaBoost algorithms and the resulting accuracy.

References

[1] T. K. Ho. Multiple classifier combination: Lessons and the next steps. In A. Kandel andH. Bunke, editors, Hybrid Methods in Pattern Recognition. World Scientific Publishing, 2002, pp. 171–198.

[2] Shipp C.A. and L.I. Kuncheva. An investigation into how AdaBoost affects classifier diversity, Proc. IPMU 2002, Annecy, France, 2002, pp. 203—208.

[3] L. Breiman. Combining predictors. In A. J. C Sharkey, editor, Combining Artificial Neural Nets, chapter 2, Springer-Verlag, 1999, pages 31-50.

[4] E. Bauer and R.Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, 1999, Machine Learning, 36:105-139.

[5] R.O. Duda, P.E.Hart, and D.G. Stork. Pattern Classification, chapter 9, John Wiley & sons, New York,2nd edition,2001, pages 453-516.

[6] R.E. Schapire. Theoretical views of boosting. In Computational Learning Theory: Fourth European Conference, EuroCOLT'99, 1999, pages 1-10.

[7] L.I. Kuncheva, combining pattern classifier wiley,USA, 2004.

[8] T. K. Ho. The random space method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8):832–844.

[9] P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. Technical Report TCD-CS-2000-02, Department of Computer Science, Trinity College, Dublin, 2000.

[10] A.Golestani and J.Azimi. A new efficient fuzzy diversity measure in classifier fusion. IADIS Conference on Applied Computing, Spain, 2006, pages 722-726.

[11] G.U. Yule. On the association of attributes in statistics. Phil. Trans. A, 194:257-319, 1900.

[12] P.H.A. Sneath and R.R. Sokal. Numerical Taxonomy. W.H. Freeman & Co., 1973.

[13] T.K Ho. The random space method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832-844, 1998.

[14] D.B. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In Proc. American Association for Artificial Intelligence, Integrating Multiple Learned Models Workshop. AAAI, 1996.

[15] G.Giancinto and F. Roli. Design of effective neural network ensembles for image classification processes. Image, Vision and Computing Journal, 2001.

[16] R. Kohavi and D.H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In L. saitta, editor, Machine Learning: Proc. 13th International Conference, Morgan Kaufman, 1996, pages 275-283.

[17] J.L. Fleiss. Statistical methods for Rates and Proportions. John Wiley & Sons, 1981.

[18] L.I. Kuncheva and C.J. Whitaker. Ten measures of diversity in classifier ensembles: limits for two classifiers. In IEEE Workshop on Intelligent Sensor Processing, Birmingham, UK, 10 2001. ISP2001.

[19] L.K. Hansen and P. Salamon. Neural network ensembles. IEEE Transaction on Pattern Analysis and Machine Intelligence, 1990, 12(10):993-1001.

[20] D. Parteidge and W.J Krzanowski. Software diversity: practical statistics for its measurement and exploitation. Information & Software Technology, 1997, 39:707-717,.

[21] D.Partridge and W.J. Krzanowski. Distinct failure diversity in multiversion software. (Personal communication 1999).

0-7695-2841-4/07 $25.00 © 2007 IEEE