Journal of Computer Science 3 (4): 195-198, 2007
ISSN 1549-3636

An Efficient Implementation of Re-Sampling Technique for High Performance Multiple Classifier Systems

[1]S.Sathiyabama,[2]K. Thyagarajah and [3]D.Ayyamuthukumar
[1]Department of MCA, K.S. Rangasamy College of Technology, Tiruchengode-637 209, India
[2]P.S.N.A..College of Engineering and Technology, Dindigul – 22, India
[3]Department of CSE, K.S. Rangasamy College of Technology, Tiruchengode-637 209, India

Abstract: Due to the large size of the database, the entire training dataset could not be used to construct the classifiers. One popular solution is to separate stream data into chunks, learn a base classifier from each chunk and then integrate all base classifiers to form Multiple classifier system (MCS). Sometimes this data streams does not include all the classes in its equal proportion as in the entire training data set. So we have newly introduced a method of Re-Sampling based on the statistical value of the class attribute. In the Proposed Method, the probability of occurrences of every class for the entire training data set have been estimated. Based on the probability, thresholds have been fixed for all the classes. When the data set have been selected randomly, the probabilities of the classes have been checked against the thresholds. The sample, which satisfies all the thresholds, is allowed to construct the Model. Otherwise, Re-sampling is performed and the process is repeated until the sample satisfies all the thresholds for the classes. The proposed method yields more accuracy than the one which does not have threshold on classes in the random samples. We have also compared the accuracy of different classifiers. Experimental results and comparative studies demonstrate the efficiency and efficacy of our method.

Key words: Accuracy, classifier, euclidean distance, sampling, threshold, normalization

INTRODUCTION

Classification has been identified as an important problem in the emerging field of data mining. While classification is a well-studied problem, only recently has there been focus on algorithms that can handle large databases. The intuition is that by classifying larger datasets, we will be able to improve the accuracy of the classification model[1,2].

In classification, there are given a set of example records, called a training set, where each record consists of several fields or attributes. Attributes are either continuous, coming from an ordered domain, or categorical, coming from an unordered domain. One of the attributes, called the classifying attribute, indicates the class to which each example belongs. The goal of classification is to assign a new object to a class from a given set of classes based on the attribute values of the object. Different methods have been proposed for the task of classification such as Decision Tree, K-Nearest Neighbor, Back Propagation Networks etc. Accuracy is an important factor in assessing the success of Classifier. It evaluates how accurately a given classifier will label the future data[3].

Literature review: Because of the large size of the database, we couldn't use the entire training data set to construct the classifier. Random sampling has been often used to handle large datasets when building a classifier[5,6].

In the literature, many of the MCSs using random sampling have been described to increase the accuracy of the classifier. Most of the combination methods used in such systems assume that classifiers forming the MCS make "independent" classification errors. This assumption is necessary to guarantee an increase of classification accuracy with respect to the accuracies provided by individual classifiers[7,8].

Recently, some researchers proposed a different approach to the development of MCSs based on the concept of Dynamic classifier selection to increase the accuracy. Roughly speaking, selection-based MCSs are based on a function that for each test pattern, dynamically select the classifier that correctly classify

Corresponding Author: S. Sathiyabama, Assis. Prof.r, Department of MCA, K.S. Rangasamy College of Technologyy, Tiruchengode-637 209, India, Tel: 9443419843, E-mail: sathyaaksrct@yahoo.com

it. The authors pointed out that selection-based MCSs, as compared with the combination-based ones, do not need of the assumption of "independent" classifiers. For each test pattern, selection-based MCSs need just one classifier that correctly classifies it[9,10,11].

Chan and Stolfo[12,13] considered partitioning the data into subsets that fit in memory and then developing a classifier on each subset in parallel. The output of multiple classifiers is combined using various algorithms to reach the final classification. Their studies showed that, the multiple classifiers system built using the random sample did not achieve better accuracy than a single classifier built using the entire training data. It is because of the class label attribute was not distributed similarly in the random sample as like in the entire training data set.

Some times, the proportion of the class attribute in the random sample is not similar to the one in the entire training data set. To the best of our knowledge, no method has considered the Probability of the class label attribute in the random samples. In our newly Proposed Method, the probability of occurrences of every class for the entire training data set have been estimated. Based on the probability of the class label attribute in the training set, thresholds have been fixed for all the classes. When the data set have been selected randomly, the probabilities of the classes have been checked against the thresholds. The sample, which satisfies all the thresholds is allowed to construct the Model. Otherwise Re-sampling performed.

Re-sampling based on threshold (RST)

Problem definition: Let us consider a classification task for M data classes 1…M. The threshold value for M classes are $T_1,T_2…T_M$. Each class is assumed to represent a set of specific patterns, each pattern being characterized by a feature vector **X**. Let us also assume that *L* different classifiers, C_j, j = 1,.,*L*, have been trained separately to solve the classification task at hand. Let $C_j(\mathbf{X}) \in \{1,…., M\}$ indicate the class label assigned to pattern **X** by the classifier C_j. In all iterations, a random sample has been selected based on the threshold value of the class label attribute.

Fixing thresholds for the classes: Let us consider the total number of samples in the training data set is N. The total number of samples in classes 1,2,,…,M is $N_1,N_2,…N_M$ respectively. Initially, the probability of occurrence of M data classes have been computed as follows:

$$P_i = \frac{N_i}{N} \quad \text{where } i = 1,2,..,M$$

In every iterations, a set of random sample R of size K has been selected. The threshold T_i is computed as:

$T_i = P_i * K * 0.75$ where i= 1,2,…M

Then the total number of training samples in each class $K_1,K_2,…,K_M$ for the random sample R is computed. If

$K_1 >= T_1$ and $K_2 >= T_2$ and….and $K_M >= T_M$

then the sample R is allowed to construct a classifier. Otherwise sample R is rejected and the process of random sample selection have been repeated until all the classifiers have been constructed.

RESULTS

To analyze the accuracy of the classifier, the earthquake data has been taken. India and adjoining regions bound by 0^oN-40^oN latitude and 65^oE-100^oE longitude are having 24637 earthquakes from the year 1000 onwards with different magnitude[4].

The earthquake data samples were collected after removing duplicates, aftershocks and earthquakes without any magnitude. Table 1 shows the details of earthquake data samples.

Table 1: Earth Quake Data Samples

Yr	Mo	Dy	Hrs	Min	Secs	Lat	Lon	Dep	Mag
1973	1	2	16	25	36	36.08	71.31	137	4.8
1973	1	2	22	25	57	31.24	88.09	33	5.2
-		-	-	-	-	--	-	-	-
-		-	-	-	-	--	-	-	-

In the earthquake data sample, a number of continuous attributes like year, month, day, hour, minute, second, latitude, longitude, depth and magnitude are there. The attribute magnitude has been considered as the class label attribute and we have categorized it in to three categories. The magnitude 7 and above is in category 1, the magnitude from 5.5 to 7 is in category 2 and the magnitude below 5.5 is in category 3. From the entire data set, 20000 data samples have been used as training data and 4637 have been used as the test data.

Our experiments have mainly aimed to:

1. Compare the accuracy of Decision Trees which is built by non RST with the one built by RST.
2. Compare the accuracy of non RST based K-Nearest Neighbor with the RST based K-Nearest Neighbor by varying K value from 1 to 25.
3. Compare the accuracy of non RST based Back Propagation Network with the RST based Back Propagation Network for the different number of hidden units from 1 to 5.
4. Compare the accuracies of Decision tree, Back Propagation Network and the K-Nearest Neighbor.

5. Compare the accuracy of combination based MCS without RST and the one with RST.

The Probability of occurrences of the classes for the entire training sample has been estimated. The category 1, 2 and 3 are having the probability 0.11%, 3.11% and 96.8% respectively. When constructing classifiers, 30% of the random sample has been chosen and the probability of occurrences of all categories of the classes have been estimated.

Decision tree: A decision tree construction process is concerned with identifying the splitting attributes and splitting criteria at every level of the tree. If samples at a node belong to two or more classes, then a test is made at the node that will result in a split. This process is recursively repeated for each of the new intermediate nodes until a completely discriminating tree is obtained[1,2]. Table 2 shows the performances of Decision tree for various set of random samples in different iterations.

K-nearest neighbor: It classifies each record in a dataset based on a combination of the classes of the k records that are most similar to it in the training dataset. The algorithm assumes that similar cases behave similarly. The Euclidean distance is used to measure the distance between two vectors. K-Nearest Neighbor(KNN) produced different Accuracies for different value of K[3].

In all iteration, the experiments have been carried out for the neighborhood size ranging from 1 to 25 for the same set of samples. The result related to the neighborhood size that provided the highest accuracy have been reported in all iterations. Table 3 shows the Accuracy of KNN for different set of samples in different iterations

Back propagation network (BPN): The Network learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label. For each training sample, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual class. These modifications are made in the "backwards" direction, that is from the output layer, through the hidden layers down to the first hidden layer[14].

To construct the Network, the data samples are normalized into the values in between 0 to 1. Initially the weights and bias have been initialized to the values in between -1 and 1. The learning rate has assumed as the value 0.9. The weights and bias have been modified during learning process[14,15].

Table 2: Accuracy of decision tree for the different random samples

Iteration	Prob. of samples			Un Ssatisfied.cat	Accu of DT
	Cat. 1	Cat. 2	Cat. 3		
1	0.06	2.51	97.43	1	65.45
2	0.02	2.29	97.69	1,2	63.12
3	0.05	3.42	96.53	1	64.23
4	0.00	2.94	97.06	1	65.76
5	0.01	3.11	96.88	1	62.68
6	0.15	3.13	96.72	2	66.34
7	0.04	2.43	97.53	1	64.92
8	0.10	2.45	97.45	Nil	68.15

Table 3: Accuracy of KNN in different iterations

Iteration	Prob. of samples			Un Ssatisfied.cat	Accu of DT
	Cat. 1	Cat. 2	Cat. 3		
1	0.05	2.93	97.02	1	76.02
2	0.07	2.22	97.71	1,2	75.61
3	0.01	2.36	97.63	1	75.01
4	0.09	2.13	97.78	2	74.95
5	0.04	2.01	97.95	1,2	77.23
6	0.03	2.98	96.99	1	76.51
7	0.06	2.41	97.53	1	77.02
8	0.01	2.67	97.32	1	75.69
9	0.09	1.92	97.99	2	76.45
10	1.12	2.49	96.39	Nil	78.67

Table 4: Accuracy of BPN for different number of hidden units

Iteration	Accuracy of MLP for different Number of Hidden units					Highest Accuracy
	1	2	3	4	5	
1	69.12	69.41	69.53	69.07	69.01	69.53
2	67.91	67.98	67.89	68.03	67.85	68.03
3	68.23	68.76	68.04	68.94	68.51	68.94
4	70.59	70.12	70.73	69.99	70.32	70.73
5	69.74	69.13	69.67	69.95	69.81	69.95
6	72.81	72.94	72.54	73.05	72.91	73.05
7	74.09	73.93	73.91	74.02	74.14	74.14

Table 5: Accuracy of BPN in different iterations

Iteration	Prob. of samples			Un Ssatisfied.cat	Accu of DT
	Cat. 1	Cat. 2	Cat. 3		
1	0.05	2.49	97.46	1	69.53
2	0.08	2.31	97.61	2	68.03
3	0.01	2.89	97.10	1	68.94
4	0.04	2.57	97.39	1	70.73
5	0.03	2.09	97.88	1,2	69.95
6	0.00	2.60	97.40	1	73.05
7	1.02	2.78	96.20	Nil	74.14

The network has trained for different number of hidden units from 1 to 5. Different accuracies have been produced by different number of hidden units for the same set of data samples. Table 4 shows the accuracy of BPN for different number of hidden units for the same set of samples. For each data set, the results related to the number of hidden units, which produced the highest accuracy have been reported. Table 5 shows the Accuracy of BPN for different set of samples in various iterations.

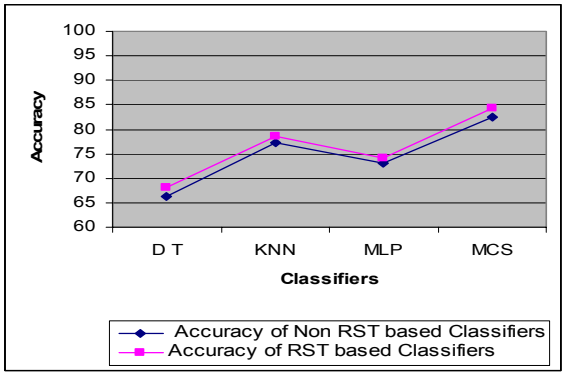

Fig. 1: Comparison of RST based classifiers with the non RST based classifiers

Combination of multiple classifier systems: The different classification techniques such as Decision tree classifier C1, K Nearest Neighbor classifier C2 and the Neural Network classifier C3 are combined using simple voting method to form MCS.

The performances of the best individual classifier which has built using non TCC based samples in each technique has been selected to form MCS and it produced 82.45% of accuracy for the test samples. The RST based classifiers in each technique have been combined using simple voting method and it has produced 84.23% accuracy. Figure 1 Compares the accuracy of RST based classifiers with the Non RST based Classifiers.

CONCLUSION

The main objective of this study was to compare the Accuracy of RST based Classifiers with the Non RST based Classifiers. Reported result showed that our Proposed RST always outperforms the other in individual model and also in combination Model.

Decision tree has relatively faster learning speed. The predictive performance of Decision Trees was not as strong as on unseen data as that obtained on the training data. The choice of neighbourhood size is always a critical problem for the KNN. We have experimented by varying the size of K from 1 to 25 and the KNN have been robust to the size of neighbourhood.

BPN has slow training time. There is always a trial and error for choosing a number of Hidden Units. So we have tried a different number of hidden units for the same set of random sample. Neural networks have high tolerance to noisy data as well as their ability to classify patterns on which they have not been trained. Neural networks could out perform other techniques because they "learn" and improve over time whereas the other techniques are static. Neural network involves long training times.

REFERENCES

1. Berson, A., S. Smith and K. Thearling, 2000. Building Data Mining Applications for CRM. Tata McGraw-Hill Edn., pp: 123-139.
2. Arun, K.P., 2001. Data Mining Techniques. University Press, pp: 153-159.
3. Ian, H.W.E.F., 1999. Data Mining–Practical Machine Learning Tools and Techniques with JAVA Implementations. Morgan kaufmann Publishers, pp: 159-169.
4. McNutt, M. and T.H. Heaton, 1981. An Evaluation of the Seismic-Window Theory for Earthquake Prediction, pp: 12-16.
5. Giorgio, G. and F. Roli, 1999. Methods for dynamic classifier selection. 10th Intl. Conf. on Image Analysis and Processing, pp: 659-664.
6. Xingquan, Z., X. Wu and Y. Yang, 2004. Dynamic classifier selection for effective mining from noisy data streams. 4th IEEE Intl. Conf. on Data Mining, pp: 305-312.
7. Xu, L., A. Krzyzak and C.Y. Suen, 1992. Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. on Systems, Man and Cyb., 22: 418-435.
8. Huang, Y.S. and C.Y. Suen, 1995. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans. on Pattern Analysis and Machine Intelligence, 17: 90-94.
9. Srihari, S.N. *et al*., 1994. Decision combination in multiple classifiers systems. IEEE Trans. on Pattern Analysis and Machine Intelligence, 16: 66-75.
10. Kittler, *et al*., 1998. On combining classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20: 226-239.
11. Woods, K. *et al*., 1997. Combination of multiple classifiers using local accuracy estimates. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19: 405-410.
12. Philip, K.C. and S.J. Stolfo, 1993. Experiments on multistrategy learning by meta learning. In Proc. 2nd Intl. Conf. on Mgmt., pp: 314-323.
13. Philip, K.C. and S.J. Stolfo 1993. Meta learning for multistrategy and parallel learning. In Proc. 2nd Intl. Workshop on Multistrategy Learning, pp: 150-165.
14. Hansen, L.K. and P. Salamon, 1990. Neural network ensembles. IEEE Trans. on PAMI, 12: 993-1001.
15. Suen, C.Y. *et al*., 1995. The combination of multiple classifiers by a neural network approach. Intl. J. Pattern Recognition and Artificial Intelligence, 9: 579-597.