A Novel Clustering-Based Ensemble Classification Model for Block Learning

Mohammad Raihanul Islam[1], Md. Mustafizur Rahman[1] and Asif Salekin[1] and Shihab Hasan Chowdhury[1] and Samiul Alam Anik[1]

[1]*Bangladesh Unversity of Engineering and Technology, Dhaka, Bangladesh*
{*raihan2108, nahidcse05, asalekin, shihab2608, samiul.anik*}*@gmail.com*

Keywords: Ensemble Classifier:K-means clustering:Artificial Neural Network

Abstract: In this paper, we have considered a real life scenario where data is available in blocks over the period of time. We have developed a dynamic cluster based ensemble of classifiers for the problem. We have applied clustering algorithm on the block of data available at that time and have trained a neural network for each of the clusters. The performance of the network is tested against the next available block of data and based on that performance the parameters of the clustering algorithm is changed at runtime. In our approach increasing the number of clusters is considered as changing of the parameter settings. The misclassified instances of the test data are also joined with the training data to refine the knowledge of the classifiers. The proposed system is capable of identifying the decision boundary of different classes based on the current block of data more precisely. An extensive experiments has been performed to evaluate this dynamic system and to compute the optimal parameters of the proposed procedure.

1 INTRODUCTION

In real world it is common phenomena that, all the learning data is not available at the same time. Data may be available in different or fixed sized small block over the period of time. The new data may contain new class or diverged data points of previous classes. We have used the term *block learning* to refer to them. In this kind of scenario it is essential that, we have to design such classification model that, it can learn from the block of data over period of time without forgetting previous experience. The *block learning* approach can be implemented by constructing an ensemble of classifier. In an ensemble classification model, multiple classifiers are trained and final decision on a test pattern is made by combining the individual decision of the classifiers on that test pattern. There is a rich description for generating ensemble classifiers are present in the literature(Rahman and Verma, 2011; Polikar, 2006; Polikar et al., 2001; Muhlbaier et al., 2001). An ensemble of classifier can be constructed by clustering. In ensemble classification, the base classifiers are generated in such a way that, they differ from each other in case of decision they made on same pattern. This process is generally known as *diversity*.

In this paper, we presented a novel method for constructing a clustering based ensemble classification model where training is performed on the block data. We have used the term *Clustering Based Ensemble Classification* (CBEC) *for Block Learning* to refer our proposed approach. We have investigated the impact of varying different clustering parameters on the overall approach. This paper is organized as follows. In Section 2 we have gone through some of the existing approaches on ensemble classification. We give the detailed description of our approaches in Section 3. We described the experimental settings and discussed our findings in Section 4. Finally we have concluded in Section 5.

2 PREVIOUS WORKS

A significant number of researches have been directed towards the construction of ensemble of classifiers. Some of the earlier works relating to this topics are presented in (Kittler et al., 1998; Hansen and Salmon, 1990; Wanas et al., 2006; Terrades et al., 2009). Different types of strategies can be initiated to construct ensemble classifiers. One of the strategy is to change the parameters of the classifiers to introduce diversity in the training process so that different decision on the same pattern can be obtained. This idea was presented in (Yamaguchi et al., 2009). Another popular approach for constructing ensemble classifiers is to manipulate the training examples. The classifiers are trained on different subsets of training examples. A general technique is known as *boosting*. Here normally training examples, which were misclassified in previous iteration are emphasized to select. AdaBoost is generalized version of boosting (Freund and Schapire, 1997). In (Polikar et al., 2001) the authors have proposed an approach for constructing ensemble of classifiers for incremental learning named Learn++. Here authors put a weight to each of the training examples based on the performance of the classifiers on previous iteration. Misclassified examples are put greater weight, so that they tend to come to the training more than the successfully classified ones. The error rates of each classifiers are calculated and the final hypothesis are selected on the basis of the that. Learn++ have a problem if the classifiers are faced with previously unseen data. The majority of the classifiers may label wrong classes on that example outvoting the classifier correctly label them. In (Muhlbaier et al., 2001) authors reported that, they developed a method where each classifier keep track of which classifiers are trained on which classes. As a result it is possible for the classifiers to manipulate the weighting of the training examples if they are not trained on them. There has been a substantial amount of research effort has been put in the the direction of evaluating the performance of the ensemble of classifiers. Researchers are concerned about the improvement of the performance and reducing bias and variance of the classifiers(Kuncheva, 2004). A detail review of the ensemble of classification can be found in (Polikar, 2006; Polikar, 2007).

3 ENSEMBLE OF CLASSIFIERS FOR BLOCK LEARNING

In this Section, we present our approach to generate ensemble of classifiers for block learning. First we describe the basic principle behind it, then we describe our method to combine the decision of the classifiers to obtain final decision on class instances.

3.1 Fundamental Principle

Our proposed approach for constructing ensemble of classifiers is based on the concept of clustering. The clustered data points tend to stay close in Euclidean space. Then if a classifier like neural network is trained on the on the each clustered data points, each of them become a *local expert* on each of the clustered data. The first step is to cluster the entire block of data points of size Λ with any clustering algorithm (we use *k-means* clustering on our approach). When the classifiers are tested, all the classifiers can give their decisions on the example. The decision of the neural networks can be combined using any fusing technique like majority voting. The misclassified instance can be incorporated into the training example. Then the updated classifier can be retrained with the combined training examples. This process will continue.

To achieve diversity, we can change the parameters of the k-means clustering algorithm. In our approach we have considered a threshold parameter θ. If accuracy of all the neural networks are below θ, then we have decided that, the current clustering can no longer show satisfactory result. In this case we have changed the parameter of the k-means clustering. In our approach, we increase the number of cluster by one. Increasing the number of cluster makes it possible that, trained neural network becomes more *locally* expert on data points.

3.2 Combining the Decision of the Classifiers

One of the major components of any ensemble system it to implement a strategy to fuse the decisions of the classifiers to reach for a final verdict. There are many combination techniques described in the literature. In our CBEC approach for Block Learning we have employed a simple version of majority voting.

Majority Voting: Consider a classification system, where there are T classifiers and C classes are available. Let be the classifiers be $1,......,T$ and the classes be $1,......,C$. Additionally assume that, the decision of the classifiers are define as $d_{t,j} \in \{0,1\}$ and if the t^{th} classifiers selects class ω_j, then $d_{t,j} = 1$, otherwise $d_{t,j} = 0$. Now in majority voting a class is ω_J if,

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} d_{t,j} \quad (1)$$

3.3 CBEC vs. Static Clustering

As mentioned earlier, we use the term *CBEC for Block Learning* to refer the approach we described in previous section. We need to compare our approach with the existing clustered ensemble approach to estimate the practicability of our approach. For static clustering, we consider that, the total data are present at one time. Then we cluster the data with k-means clustering with $k = \kappa$. Then we train κ neural networks with the data of κ clusters. After that we classify each test example with that neural networks whose center of cluster is nearest that test example. We run this procedure from $\kappa = 1$ to $maxCluster$. We select the cluster number $\kappa = desiredCluster$ for which the error rate is lowest. For CBEC method, we assume that, the entire data are not present at the beginning. We have partitioned the whole data in to equal blocks to implement our method. Then we apply k-means clustering on the first block and execute the procedure describe previously on the blocks of data. At the end of the procedure, the number cluster from CBEC and the value of $desiredCluster$ is compared. We present the algorithmic procedure of this method in Algorithm 1.

Algorithm 1 CBEC for Block Learning

initialize threshold = θ
while there is one more data block available **do**
    build training set
    apply k-means clustering with $k = \kappa$
    obtain clustered data points $cluster_1, cluster_2 ......, cluster_\kappa$ from k-means clustering
    for all i=1 to κ **do**
        train neural network nn_i from the clustered data points $cluster_i$
        test neural network nn_i with the next available block of data
        calculate error rate e_i from testing
    end for
    combine all the decisions for the neural networks to obtain final decision for each test example
    calculate and update the accuracy for the cluster κ
    join the misclassified instance with the test examples
    if for all the error rate ($e_1, e_2, ...e\kappa$) is $> \theta$ **then**
        $\kappa \leftarrow \kappa + 1$
    end if
end while
compare the number cluster generated in CBEC and static procedure
report the accuracy of each cluster

4 EXPERIMENTS AND RESULTS

We have experimented our approach on a number of empirical data sets to validate our proposed method. The proposed ensemble classification model has been tested on various dataset form UCI Machine Learning Repository. The simulation of the proposed approach has been performed in a PC with processor INTEL core $i3$, 4GB RAM. The simulation has been implemented in Matlab 2012. The k-means algorithms were implemented with different distance metric namely Euclidean distance, Absolute Distance and Cosine Similarity. K-means clustering starts with k random initial points and the number of iteration for clustering was set to 5. We have used multi-layer neural networks with back-propagation learning for training the clustered data. The number of neurons in hidden layer was 10 and the number of iteration for training was 1000. We have also varied the size of the data block with size of 1/10 , 1/20 and 1/40 of the total data. Last the threshold θ have also been altered with value 0.1, 0.2, 0.3 and 0.5. The whole process has been simulated $C = 10$ times. The optimal parameter for the experimented data set has been computed. The cluster number which occurred most in each of the data set are computed and reported in Table 1.

Dataset Name	Distance	Size	Threshold	Optimal Cluster
Glass	Euclidean	1/20	0.3	5
OCR	Absolute	1/10	0.2	2
Breast Cancer	Euclidean, Absolute	1/10	0.2	3
Wine	Euclidean	1/10	0.2	6
Pendigit	Euclidean, Absolute, Cosine	1/10	0.1	2
Satellite	Euclidean, Absolute	1/10, 1/40	0.2	2
Spam	Euclidean, Absolute	1/10	0.1	2

Table 1: Optimal Parameters for Experimented Datasets

We have shown the frequency distribution of clusters of the CBEC procedure and static clusters for different datasets in Figure 1 in percentage (%). In the simulation the number of cluster generated at the end of the each process is shown in Figure 1 in percentage.

5 CONCLUSION

In this paper, we present a novel approach for generating ensemble of classifiers which are trained from blocks of data over the period of time. In our proposed methods we conjecture that, in a clustered ensemble classification model if remote data points and unseen classes arrive at the feature space increasing the number of clusters the classifiers are expected to learn the

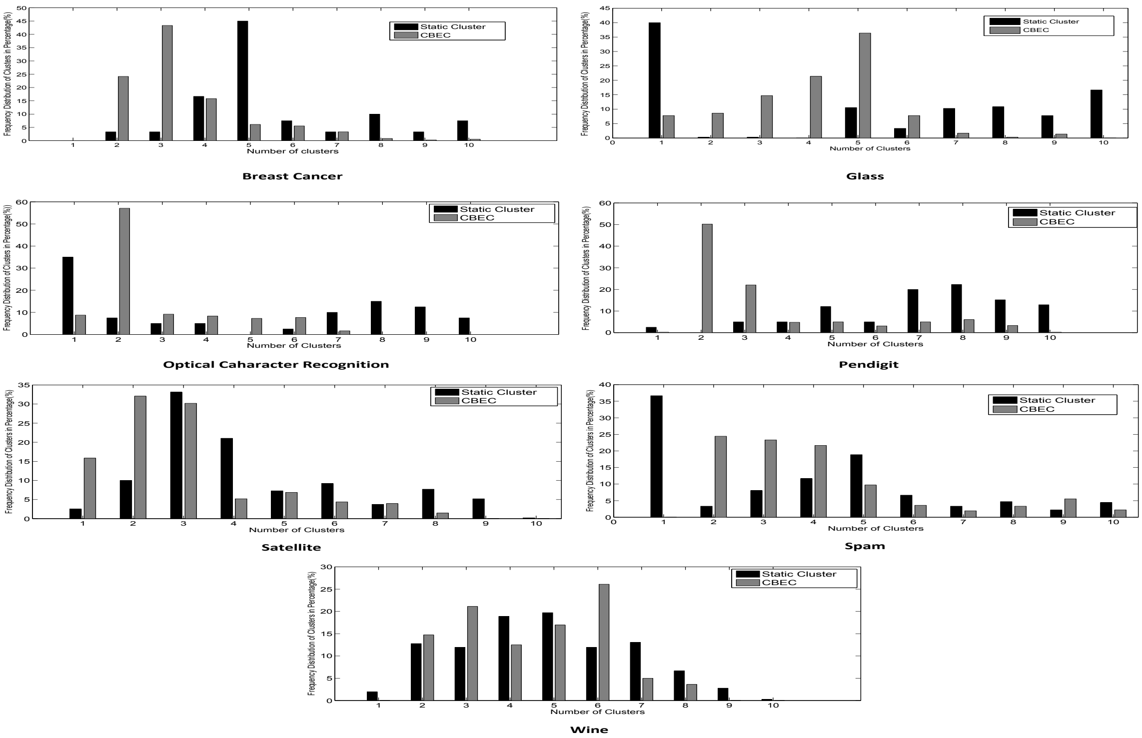

Figure 1: Frequency Distribution of Cluster for Different Dataset

decision boundary more precisely and each become more expert on the respective cluster more locally. We have performed extensive analysis of our approached. We believe our approach is highly suitable for various practical environments where data are coming in different time like financial data, satellite image. Moreover we are further interested in generating ensemble of clustering for dynamic environment.

REFERENCES

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Hansen, L. K. and Salmon, P. (1990). Neural network ensembles. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.

Muhlbaier, M. D., Topalis, A., and Polikar, R. (2001). Learn++.NC: Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes. *IEEE Transactions on Neural Networks*, 20(1):152–167.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

Polikar, R. (2007). Bootstrap—Inspired techniques in computation intelligence. *IEEE Signal Processing Magazine*, 24(4):59–72.

Polikar, R., Udpa, L., Udpa, S. S., and Honavar, V. (2001). Learn++: An Incremental Learning Algorithm for Supervised Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(4):497–508.

Rahman, A. and Verma, B. (2011). Novel Layered Clustering-Based Approach for Generating Ensemble of Classifiers. *IEEE Transactions on Neural Networks*, 22(5):781–792.

Terrades, O. R., Valveny, E., and Tabbone, S. (2009). Optimal Classifier Fusion in a Non-Bayesian Probabilistic Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1630–1644.

Wanas, N. M., Dara, R. A., and Kamel, M. S. (2006). Adaptive fusion and cooperative training for classifier ensembles. *Pattern Recognition*, 39(9):1781–1794.

Yamaguchi, T., Mackin, K. J., Nunohiro, E., Park, J. G., Hara, K., Matsushita, K., Ohshiro, M., and Yamasaki, K. (2009). Artificial neural network ensemble-based land-cover classifiers using MODIS data. *Artificial Life and Robotics*, 13(2):570–574.