ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 7, July 2016

Cluster based boosting for high dimensional data

Rutuja Shirbhate , Dr. S. D. Babar

***Abstract* -Data Dimensionality is crucial for learning and prediction systems. Term Curse of High Dimensionality means when data becomes more dimensional, complexity in learning increases. CBB may face an issue Accuracy degradation due to irrelevant features. Our objective is to use a feature selection technique which will remove irrelevant features and redundancies from the available dataset which will reduce the impact of high dimensional data solve the curse of dimensionality issue. This disadvantage of boosting is recover by cluster based boosting in which data is clustered before boosting and depend on the cluster boosting is performed. In CBB all types of features data is used for clustering. Due to consideration of irrelevant feature there is possibility of wrong clustering. Wrong clustered data may result into negative affect on boosting performance. Feature selection is applied before clustering on training data to overcome this problem. Use of boosting in many applications proved its effectiveness. Although its success, boosting had some issues.**

***Keywords* - Feature Selection, Boosting, Clustering**.

I. INTRODUCTION

Classifiers in the data mining can be divided by their learning process or representation of extracted knowledge. support vector machine (SVM), decision trees like ID3, C4.5, k-nearest neighbor classifiers, and Probability based classifiers like Naive Bayes. Boosting means, once learning process is completed and classifier is learned, boosting generates subsequent classifiers by learning incorrect predicted examples by previous classifier. All generated classifiers then used for classification of the test data. Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in learning.

CBB clusters are created using all features in the data, this works well on standard data set. But in real world dataset contains large number of feature and may result into inaccurate clusters. Such inaccurate clusters can affect CBB negatively.

To fix this problem there is need to apply feature selection and use data of selected features for clustering. But in a high dimensional feature space cluster based boosting may face performance degradation and result inaccuracy due to redundancy and irrelevancy of the features. So our focus is on improving the accuracy of CBB even in case of high dimensional data. Despite success of boosting, there were some disadvantages in the boosting process. Boosting could not deal with the noisy data and troublesome areas in the training data. Such noisy data or data with troublesome area cannot handle by boosting.

To overcome above issues paper [6] proposed Cluster-based boosting (CBB) method which can deal with the noisy data and troublesome areas in the training data. Such inaccurate clusters can affect CBB negatively. To fix this problem there is need to apply feature selection and use data of selected features for clustering. Our system proposes extension to existing CBB. Feature selection will be used in CBB in proposed system.

II. LITERATURE SURVEY

1]As described in this paper [1], for getting better results, need to improve distraction process to accomplish accuracy and clustering quality. In this paper proposed a strong and hardy multi clustering solution which is based on general proposition of boosting by boosting a simple clustering algorithm. Create multiple sets of clusters and combines it into final set of cluster using algorithm and weighted vote. Results shows on improvement of quality of clustering using boosting by boosting.

[2] Boosting process works on untrue classified instances. This paper implemented Breiman’s arc-gv algorithm for maximizing margins also it explains why boosting is resistant to over fitting and how it refines the decision boundary for accurate predictions. Margin is a prediction of accuracy, proves predictive accuracy improves with number of boosting iterations.

[3] When less noisy data is present AdaBoost rarely suffers the issues of over fitting. The adaptive boosting algorithm known as AdaBoost provided great success and proved as important developments in classification methodologies Over fitting problem occurs in Adaboost when rate of noise data is high. To improve the strong and hardyness of AdaBoost, paper proposed two officially accepted schemes from the standpoint of mathematical programming. These two algorithms AdaBoostKL and AdaBoostNorm2 are proposed based on the different penalty functions .The performance of AdaBoostKL is considered as a best as compare to all among AdaBoost algorithm.

[4] On some dataset generally boosting faces over-fitting issues and works well on some another datasets. Authors of

this paper describes that this problem happens due to presence of overlapping classes. To overcome this issue boosting, ‘confusing samples’ are evaluated using Bayesian classifier and removed during boosting phase. In this paper, performing analysis of Adaboost without confusing examples . instances which are misclassified this classifier are considered as confusing instances. AdaBoost algorithm is used for boosting purpose. Using this boosting process confusing instances are removed which results of the experiments proved that observation about overlapping

classes was correct.

[5]Ensembles of classifiers are obtained by generating and combining base classifiers, constructed using other machine learning methods. To increase the predictive accuracy with respect to the base classifiers. For creating ensembles, boosting is used and AdaBoost is the most prominent . General approach for improving classifier performances is boosting. Boosting is a best method in the machine learning community for improving the performance of any learning algorithm. Boosting is a process than unite the all weak classifiers that gives wrong predictive accuracy to strong classifier. In this paper, Ganatra explains prehensive evolution and evolution of boosting on various criteria with Bagging. Experiments showed that prediction accuracy of boosting is most better than bagging which classified the samples more correctly.

[6]In This paper, drawbacks of boosting is overcome. paper consider two drawbacks, which includes , boosting uses wrongly predicted data for subsequent function learning and second, instances where their relevant features are different from the rest of the training data. The process that partitions the training data into clusters and then integrates these clusters directly in boosting process. The experiments shows that increase in the performance of boosting.

III. PROPOSED SYSTEM

Proposed system is extension of CBB. Feature selection process is added to select most relevant features and removed irrelevant features from set of all features. Feature selection works in following steps.

3.1 Irrelevant feature removal

Symmetric uncertainty (SU) of each feature with class variable is calculated using

SU (X, Y) = 2 * Gain (X|Y) / H(X) +H(Y) (1)

Where(X) is entropy of discrete random variable X.

H(X) = $\sum_{x \in X} p(x) \log p(x)$ (2)

Where p(x) is prior probability for all values of X

Gain (X|Y) = H(X) – H (X|Y)

Where X is feature and Y is class feature.

If SU of feature is less than threshold value then this feature is considered as irrelevant feature. Next procedure is performed to remove redundant features from the feature set.

3.2 Minimal Spanning Tree creation

Symmetric uncertainty (SU) of each feature with each other feature is calculated. Directed graph G (V, E) is generated where V is set of features and E is set of edges where each edge eij done SU values between nodes. Once Graph is generated, Minimal Spanning tree is generated using Prims algorithm.

3.3 Feature selection

For each edge in the MST, following criteria is checked

If SU (Fi,Fj) < SU(Fi,C) ^ SU(Fi,Fj) < SU(Fj,C) then remove Eij

If edge satisfies the criteria then that edge is removed from the MST. After above process nodes of the remaining edges are selected features. Data of the selected features is used for clustering and learning purpose as shown in figure. All further process is same as CBB discussed as follows.

In the past boosting only considers incorrectly predicted data for learning of subsequent functions this was the cause it cannot handle the noisy data and data with troublesome area. To deal with these two problems of boosting, CBB used both correctly predicted and incorrectly predicted data for training of the subsequent functions.

First the database is clustered into k clusters. . K is varied from 2 to m, for each k separated cluster set is generated. For each cluster set summation of BIC of member cluster is taken ass^k, cluster set π^r

With minimum s^k is considered for further process of boosting. During cluster set selection process, Initial function/ classifier is trained on D using supervised learning system which will yield First function in Function set F. First function will be evaluated using data in the selected cluster set.

After the evaluation, type of each cluster is determined based on the results given by the initial function and the label of the data in the cluster. Cluster can be of four types- Homogenous Prospering (HOP), Homogenous Struggling (HOS), Heterogeneous Prospering (HEP) and Heterogeneous Struggling (HES).Cluster is Homogenous if it contains all instances with the same labels otherwise it is Heterogeneous. Cluster is prospering if all instances in the cluster are predicted correctly otherwise it is struggling. Depend on the boosting or single function is learned on cluster data. If the cluster is HEP or HES then boosting is performed with learning rate 1, 0.5 respectively. If cluster is HOS then single function is learned using supervised learning algorithm. No subsequent function learned is cluster type is HOP. Function set F is used for further testing.

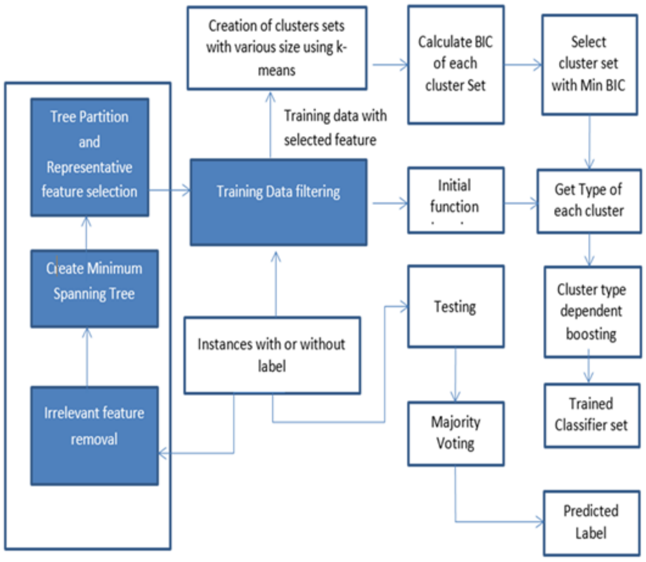
Creation of clusters sets with various size using k-means
Calculate BIC of each cluster Set
Select cluster set with Min BIC
Tree Partition and Representative feature selection
Training data with selected feature
Training Data filtering
Initial function
Get Type of each cluster
Create Minimum Spanning Tree
Testing
Cluster type dependent boosting
Instances with or without label
Irrelevant feature removal
Majority Voting
Trained Classifier set
Predicted Label

Fig. 1 Architecture of Proposed System

IV. MATHEMATICAL MODEL

Input Set {I1, I2, I3, I4, I5, I6, I7,I8, I9, I10, I11}

I1: Feature dataset

I2: Symmetric uncertainty value of each feature

I3: Redundant relevant features

I4: Graph (V, E)

I5: Minimum spanning tree

I6: Training data of relevant features

I7: Supervised learning algorithm.

I8: Produced cluster set.

I9: Cluster set with associated BIC.

I10: Selected cluster set with lowest BIC.

I11: Learned clusters based on their type

Process Set {P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14}

P1: Symmetric uncertainty (SU) of each feature with class variable is calculated using

SU (X, Y) = 2 * Gain (X|Y) / H(X) +H(Y)

Where H(X) is entropy of discrete random variable X

$H(X) = \sum_{x\,\epsilon X} p(x) \log p(x)$

Where p(x) is prior probability for all values of X

Gain (X|Y) = H(X) – H (X|Y)

Where X is feature and Y is class feature.

P2: Compare it with the threshold value, if SU is less than threshold consider the feature as irrelevant feature

P3: To remove redundancy, generate the graph (V, E) using features.

P4: Creation of Minimum spanning tree using prims algorithm.

P5: For each edge in the minimum spanning tree, following criteria is checked

If SU (Fi,Fj) < SU(Fi,C) ^ SU(Fi,Fj) < SU(Fj,C)

Then remove Eij

Remaining edges in tree considered as relevant features.

P6: Removing irrelevant and redundant features using feature selection technique.

P7: K-means clustering on I6 with following objective

$$\sum_{c=1}^{k} \sum_{Xi\in\pi c} \|Xi - Mc\|^2 \qquad (1)$$

Where xi is the instance, pc is the cluster, mc is the cluster centroid, and norm squared is the distance between the member instance and the cluster center

P8: Find out BIC for each cluster set produced in the P7

$$\text{BIC}\ (\pi c) \quad = |x| \ln \sigma^2 \ + k \ln |x| \qquad (2)$$

Where x is all the training data in cluster pc and σ^2 is the same as the inner summation in (1).

P9: Get Cluster set with minimum BIC

P10: Supervised learning System learn (D, S)

P11: get Type of the cluster

P12: Learn multiple functions using eta learning rate depends on the type of the cluster

P13: Vote for learned functions from boosting process

P14: Use Weighted function for testing of the data

Output Set: {O1, O2, O3, O4, O5, O6, O7, O8, O9, O10, O11, O12, O13}

O1: Symmetric uncertainty value of each feature

O2: Feature set without irrelevant features but with redundancy

O3: Graph (V, E)

O4: Minimum spanning tree

O5: Training data containing relevant features

O6: Cluster set with k number of clusters

O7: Set of BIC value of each cluster set

O8: Cluster set with Minimum BIC value

O9: Classifier formed by P10

O10: Type of the cluster

O11: Set of multiple classifiers formed in Boosting process P10

O12: Weight to each function in O10

O13: Predicted class of the test instance

V.RESULTS AND ANALYSIS

The experimental evaluation shows the effect of applying feature selection technique before applying cluster based boosting and also shows the time and memory requirements for existing and for proposed technique. The proposed method improves the accuracy when the data is high dimensional as irrelevant and redundant features are removed from it, the KDD dataset is used because this dataset is high dimensional.

For experiment KDD dataset with 1500 instances is used for training. Results showed that the CBB takes 4.46 seconds time and 16.91 MB memory to complete a task and the proposed method takes 0.48 seconds time and 14.87 MB memory respectively.

Table 1 CBB with Feature Selection vs CBB without feature selection

Parameters	CBB with Feature Selection	CBB without feature selection
Accuracy	1.0	0.99
Memory	13.78 MB	18.72 MB
Time	0.48 sec	4.46 sec

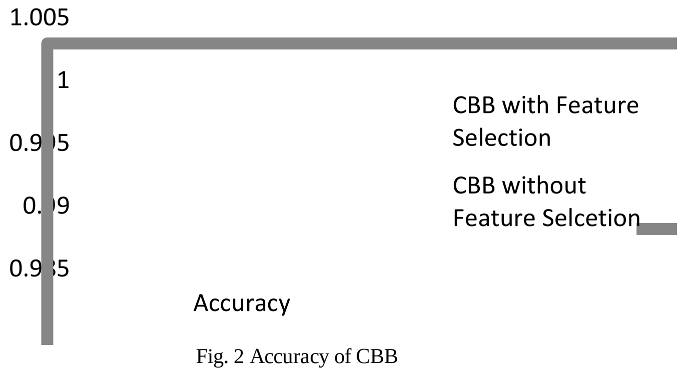

Fig. 2 Accuracy of CBB

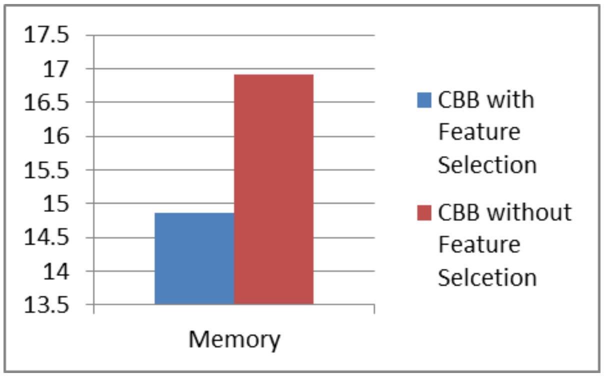

Fig. 3 Memory of CBB

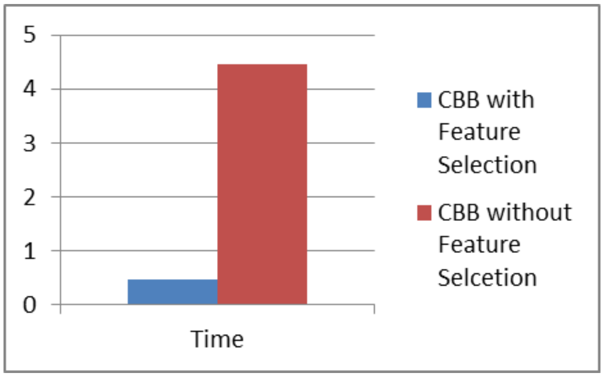

Fig. 4 Time of CBB

VI. CONCLUSION

In this paper, we discussed and explained various boosting problem and proposed solutions and also described some clustering techniques. Boosting proved advantageous for more accurate results in machine learning. Cluster based boosting approach addresses limitations in boosting on supervised learning algorithms. In CBB clusters are created using all features in the data, this works well on standard data set. But in real world dataset contains large number of feature and may result into inaccurate clusters. Such inaccurate clusters can affect CBB negatively. To address this problem we applied feature selection technique before boosting.

REFERENCES

[1] D. Frossyniotis, A. Likas, and A. Stafylopatis, "A clustering method based on boosting," Pattern Recog. Lett., vol. 25, pp. 641–654, 2004.

[2] L. Reyzin and R. Schapire, "How boosting the margin can alsoboost classifier complexity," in Proc. Int. Conf. Mach. Learn., 2006,pp. 753–760.

[3] Y. Sun, J. Li, and W. Hager, "Two new regularized adaboost algorithms,"in Proc. Int. Conf. Mach. Learn. Appl., 2004, pp. 41–48.

[4] A. Vezhnevets and O. Barinova, "Avoiding boosting overfitting by removing confusing samples," in Proc. Eur. Conf. Mach. Learn.,2007, pp. 430–441.

[5] A. Ganatra and Y. Kosta, "Comprehensive evolution and evaluationof boosting," Int. J. Comput. Theory Eng., vol. 2, pp. 931–936,2010.

[6] "Cluster-Based Boosting,"L. Dee Miller and Leen-Kiat Soh, Member, IEEE

[7] http://www.stoimen.com/blog/2012/11/19/computer-algorithms-prims-minimum-spanning-tree/

Rutuja Shirbhate completed bachelor degree of engineering from Sant Gadge Baba Amravati University, Maharashtra, India, in 2012.Currently pursuing masters in computer networks from savitribai phule pune university, Maharashtra India.

Dr.S.D.Babar is ISTE life member. He is graduated in computer engineering from Pune University, Maharashtra, India, in 2002 and received master in computer engineering from Pune University, Maharashtra, India, in 2006. From 2002 to 2003, he was working as lecturer in D.Y.Patil college of engineering, pune, India. From 2005 to 2006, he was working as lecturer in Rajashri Shahu College of engineering, Pune. From July 2006, he has been working as an assistant professor in department of information technology ,STES's, lonavala, India. He has PH.D in wireless communication at center for TeleInFrastruktur (CTIF), Aalborg university, Denmark. Currently he is working as HOD of Computer Engineering, Sinhgad Institute of Technology, Lonavala, Pune, Maharashtra, India.