

Enhanced Cluster Based Boosting in High Dimensional Data

P. Velvizhy, S. Abayambigai, and A. Kannan

Department of Computer Science and Engineering, Anna University, Chennai, India

Copyright © 2016 ISSR Journals. This is an open access article distributed under the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: Boosting is the general method which converts any weak learning algorithm into strong learner in order to improve the accuracy. The limitations in boosting is overfitting on the training data and filtering out the correct data in the subsequent function since boosting concentrates on regions not predicted well by other learners. So, the cluster based boosting (CBB) approach is used to address limitations in boosting. In this paper, initially X-Means algorithm is used to cluster the data and the clusters are selectively boosted based on the additional structure information provided by clusters and previous function accuracy on the member data. To apply Cluster Based Boosting to the high dimensional data, dimensionality reduction technique is performed. In this paper, we apply Global Redundancy Minimization frame work which considers the redundancy of the feature with all other features. The selected features will contribute more mutual information for prediction. This frame work can be used with any other feature selection technique. We provide experimental results on various dataset. These results demonstrate the effectiveness of Global redundancy framework and also effectiveness of Cluster Based Boosting with Global redundancy minimization framework than classifier with global redundancy minimization framework.

KEYWORDS: Boosting, Clustering algorithms, Margin theory, Machine learning, feature selection, redundancy minimization.

1 INTRODUCTION

Boosting is a method to improve the accuracy of any given machine learning algorithm. For supervised learners, to find a highly accurate prediction rule is certainly a difficult task. To overcome this, boosting is used. Since boosting is an iterative process and each time when it calls the base learning algorithm, it generates a new weak prediction rule, and after several rounds, the boosting algorithm must combine these weak rules into a strong prediction rule. This rule will be much more accurate than any one of the weak rules generated [1], [2].

In spite of its success, boosting such as adaboost has various problems [3]. They are (1) Training data contains label noise - where the labels of the instances provided are actually wrong (2) Training data with troublesome areas- where the relevant features of the instances are different from the rest of the training data and (3) Filtering out the data in subsequent functions when the training data contains troublesome areas and/or label noise.

In boosting, every learner covers the complete training data. The learners are dependent of one another and are biased towards the data that are wrongly predicted in the previous iteration. When the training data contains label noise then the boosting learning function fails to learn correctly. Though initial function is correct, boosting does not realize that the labels were wrong. Thus, boosting focuses subsequent functions on learning how to “correctly” predict these instances assuming that the wrong labels provided are correct [4], [5].

In boosting, overfitting occurs due to overlapping regions. The samples that are located in the overlapping region are more likely to be misclassified. Adaboost works by increasing the weight of samples that are misclassified in the previous iterations [2]. In our work we regard the overlapping regions as heterogeneous.

Another problem with boosting is due to the way it learns the subsequent function. Boosting works by filtering out some correctly classified instances and withheld the incorrect instances in the subsequent iterations. This can result in complexity

and higher probability of overfitting because some correctly predicted instances may be similar to the incorrectly classified instances in the heterogeneous region [6].

To address the limitations of boosting, we propose a novel approach called cluster based boosting (CBB). CBB uses clusters to partition data and then the boosting is applied on the clusters containing highly similar data which helps to reduce the complexity and to mitigate the overfitting [6].

To apply cluster based boosting to high dimensional data, dimensionality reduction is performed on the original data. The microarray and gene expression data suffers curse of dimensionality issue where the number of samples is much less than the number of feature which leads to overfitting of data.

In feature extraction two or more features combined to form a new feature. When the features have natural meaning they cannot be combined. Hence feature selection becomes important.

In feature ranking correlated features tend to get similar rankings, as they are considered equally significant for classification. As a result, the top ranked features obtained are often correlated to each other. From statistics point of view, these correlated features provide redundant information and may not introduce any extra useful information.

In mRMR, redundancy between selected features is considered. However, it is not optimal because it considers only the local redundancy but not the global redundancy [7], [8].

In this work, we use Global Redundancy Minimization (GRM) framework [9] which considers the redundancy of a feature with all other feature. It globally minimizes feature redundancy with maximizing the ranking scores. The selected features will contribute more information to the latent variable and minimum global redundancy. The feature ranking results of any supervised or unsupervised feature selection method can be used as the input of GRM framework.

We evaluate GRM using two studies. First, we compare feature selection with GRM based feature selection in classifiers. Second, we compare feature selection with GRM based feature selection in Cluster Based Boosting technique.

The rest of this paper is organized as follows. Section 1 provides the background on feature selection, boosting and related work on using clustering and boosting. Section 2 provides a more in-depth discussion on the boosting problems, how our CBB approach addresses these problems and CBB in high dimensional data. Section 3 provides the experimental setup and discusses the results from our studies. Section 4 concludes and discusses future work.

2 LITERATURE SURVEY

In this work, we provide the related works on boosting and margin theory and also the combination of boosting and clustering. The two methods of boosting are boosting by resampling and boosting by reweighting. Boosting by reweighting can be applied only to supervised learners that are designed to use instance weights while boosting by reweighting can be applied to all supervised learners. Hence boosting by resampling is much more appropriate. A new training dataset same as size of original dataset is created where the probability that each instance is selected is directly proportional to its assigned weights [10].

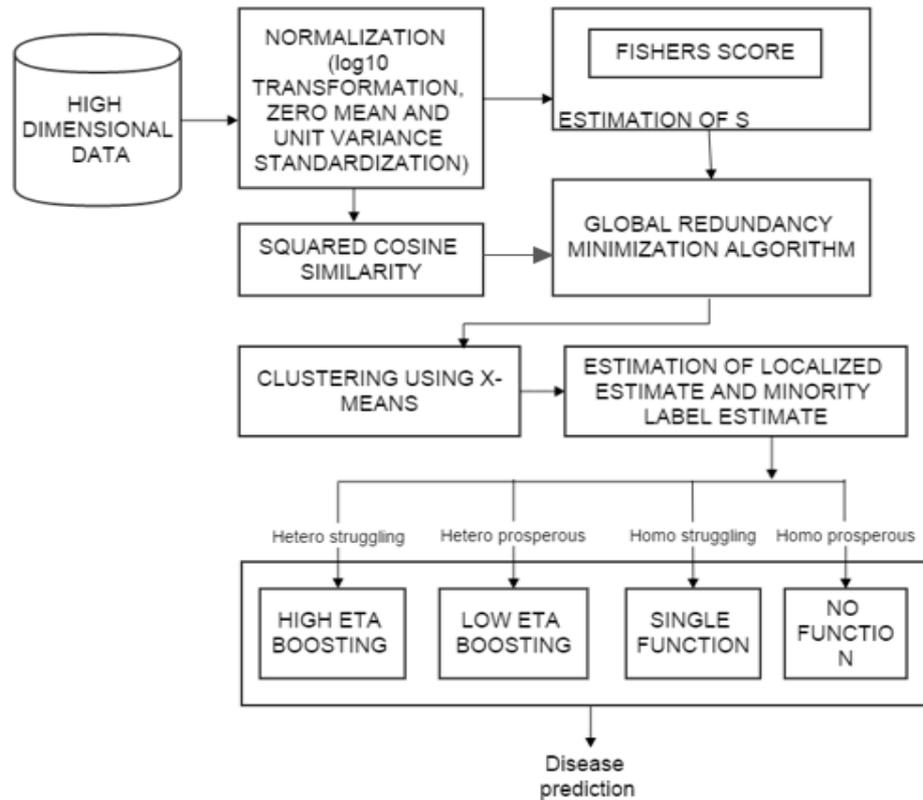


Fig. 1. System Architecture

2.1 BOOSTING AND MARGIN THEORY

In margin theory, Schapire et al. [11] made an attempt to explain boosting by considering the margin. The margin of an instance is a good indicator of classification confidence and confidence in the prediction of multiple functions. The margin on a single data instance depends on the weighted votes for multiple functions. In turn, the magnitude of the margin represents the strength of agreement between those functions and the confidence of the final decision boundary. Using these margins, it is possible to prove that predictive accuracy continues to increase with the number of boosting iterations explaining resistance to overfitting. Further extensions to margin theory have examined how the margin distribution (including margin average and variance) is connected to the predictive accuracy^[12]. The authors show how, by learning additional subsequent functions, boosting continues to improve the margin resulting in a more refined decision boundary (with higher predictive accuracy).

2.2 COMBINATION OF CLUSTERING AND BOOSTING

Here we provide related works on the combination of clustering and boosting. The three ways are (1) Boosting is used to improve clustering, (2) Boosting and clustering to improve any given supervised machine algorithm and (3) Clustering to improve boosting which is similar to our work.

There has been very little work on how clustering improves boosting. In kim et al [13] uses k-means clustering algorithm to address the label noise problem in boosting. K-Means clustering algorithm is applied to training data and each clusters is compared with the cluster in the opposite class using the mahalanobis distance. When the clusters belongs to different clusters are close enough they are pruned. Then, boosting is done on the remaining training data. We refer this as prune boost algorithm. This algorithm does not explain about various issues such as subsequent function ignoring troublesome areas and subsequent function that are complex. Cluster Based Boosting technique addresses this issue. To apply Cluster Based Boosting technique in high dimensional data, feature selection should be done.

2.3 FEATURE SELECTION

Feature selection is to select informative and relevant features from the high-dimensional space. It plays a crucial role in many practical applications. Feature selection can speed up the learning process and decrease the algorithm running time in the real applications.

Based on the training set, feature selection algorithms can be categorized into supervised, semi-supervised and unsupervised. Again, supervised feature selection methods can further be categorized into filter models, embedded models and wrapper models.

The filter model [14] separates feature selection from classifier so that the bias of a learner does not interact with the bias of a feature selection algorithm. Relief, Fisher score and Information Gain based methods are some of the frequently used algorithms of the filter model.

The wrapper model [15] uses the predictive accuracy of a predetermined learner to determine the quality of selected features. The wrapper method works better than the filter method but it is expensive than filter model and is not suitable for the large scale data. To bridge this gap embedded method is introduced.

The top ranked features are correlated and they will not introduce any extra useful information. In previous research, Ding et al. recognized this issue and proposed the minimum Redundancy Maximum Relevance Feature Selection (mRMR) model [7] [8] to minimize the redundancy. However it is greedy search and the global redundancy is not considered. To address this issue Global Redundancy Minimization framework is used which considers redundancy with all other features.

3 METHODOLOGY

In this section we first provide a more detailed discussion on boosting problem. Then we discuss our cluster based solution.

3.1 ISSUES IN BOOSTING

Here we further provide various issues on the boosting such as filtering out correct instances in subsequent functions and label noise.

3.1.1 FILTERING IN SUBSEQUENT FUNCTION

The first limitation of boosting is filtering the correctly predicted instance that is required to learn actual decision boundary which makes them unavailable for subsequent functions. Such filtering becomes problematic when training data contains troublesome areas.

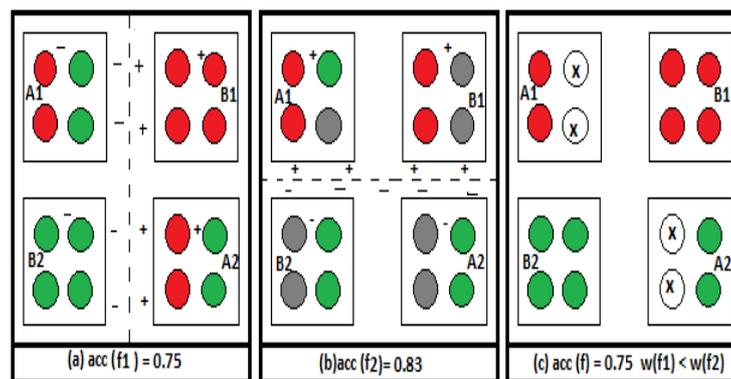


Fig. 2. Example of how filtering affects subsequent function given troublesome (A1 and A2) and normal (B1 and B2) areas. (a) After learning the initial function (b) after learning a subsequent function, and (c) boosting prediction

To illustrate this phenomenon Fig. 2 provides an example how filtering affects subsequent function. Areas A1 and A2 are troublesome areas. The red shaded region represents the positive instance and green shaded represents the negative instance and grey shaded regions are missed out samples while the x encircled are wrongly classified instance. In fig.2a vertical line represents the decision boundary which partitions the positive and negative class. This decision boundary

correctly predicts some instances in the troublesome area. In fig. 2b consider the grey shaded regions as missed out samples. These grey shaded regions are correctly predicted instance which is filtered in the subsequent function. Due to this there is change in the decision boundary. At first this seems to be working fine because the $acc(f_1)$ is less than the $acc(f_2)$ but f_2 fails to learn properly due to missed out samples.

Fig. 2c shows the weighted vote from both functions to predict the final label. f_2 gets higher weightage than f_1 because of higher accuracy achieved. The final decision boundary predicts the correct label for all the instances in areas B1 and B2 but is wrong on half of the instances in troublesome regions. The boosting has difficulty because neither function learns about troublesome areas.

3.1.2 OVERFITTING IN SUBSEQUENT FUNCTION

Learning subsequent functions on widely separated instances requires complex functions. These functions have higher accuracy and results in overfitting of subsequent function. The function with higher accuracy gets a higher weightage. This overfitting is propagated final decision since those overfitted function will have higher accuracy and higher weightage [16].

3.2 CLUSTER BASED BOOSTING IN HIGH DIMENSIONAL DATA

Here we further discuss our cluster based boosting solution in high dimensional data. First the feature selection is performed on the training data. In GRM framework, the squared cosine similarity matrix is constructed. Then the feature score s is computed using any supervised feature selection algorithm. The refined feature score z is obtained by applying the GRM framework [9] solving the objective function (1). Now the top k features are selected.

$$\text{Min } z^T A z \quad \text{where } z^T \mathbf{1} = 1; z \geq 0 \tag{1}$$

The function $z^T A z$ represents the global feature redundancy in the refined feature ranking. z^T represents the consistency between the refined feature score z and the original score s . The correlation matrix A must be positive semidefinite. If it is not positive semidefinite then it leads to nonconvex problem and global optimum cannot be achieved. In contrast the positive semidefinite is guaranteed by squared cosine similarity.

Now the selected training data is decomposed into clusters which encapsulate highly similar instances in same cluster. X-Means, the modified version of k-means [17] is used for clustering. The goal of x-means is to assign each instance to each cluster that minimizes following function. X-Means learns the appropriate number of clusters automatically [18] by increasing the cluster count till it achieves lowest Bayesian information criterion (BIC) [19] value.

$$\text{BIC}(\pi_c) = |x| \ln \sigma^2 + k \ln |x| \tag{2}$$

After clustering, CBB performs selective boosting based on the cluster type. After selective boosting, the set of functions is assigned the weighted vote and used to predict the labels for a new instance. Weighted vote is calculated using the following function [20].

$$\text{Vote}(f_t) = \eta \ln((1-\epsilon) / \epsilon) \tag{3}$$

The cluster type is computed using the localized estimate metric and the minority label estimate (MLE) metric. The localized estimate metric is used to find whether the cluster is struggling or prosperous.

$$\text{Localized Estimate}(\pi_c) = \begin{cases} \text{Prospering if } acc(F, \pi_c) \geq 1 - \delta_1 \\ \text{Struggling, Otherwise} \end{cases} \tag{4}$$

Where $acc(F, \pi_c)$ is the accuracy of the previous function evaluated on the cluster members. δ_1 is the tunable parameter and the value ranges from 0.1 to 0.3. The minority label estimate is used to find whether the cluster is heterogeneous or homogeneous.

$$\text{Minority Label Estimate}(\pi_c) = \begin{cases} \text{Homogeneous if } \text{minority}(\pi_c) < \delta_2 \\ \text{Heterogeneous, Otherwise} \end{cases} \tag{5}$$

Where $\text{minority}(\pi_c)$ is the minority label percentage on the cluster members and δ_2 is the tunable parameter and the value ranges from 0.2 to 0.4. If the cluster is Heterogeneous Struggling (HES), high-eta boosting has a learning rate for AdaBoost ($\eta=1$). Otherwise, if the cluster is Heterogeneous Prospering (HEP), low-eta boosting has a learning rate for AdaBoost ($\eta=.5$). Otherwise, if the cluster is Homogeneous Struggling (HES), a single function is learned without boosting. No functions are learned if the cluster is Homogeneous Prospering (HEP) to avoid learning label noise.

4 IMPLEMENTATION

4.1 DATASET DESCRIPTION

LUNG data set contains 203 samples. These samples are from five classes, which have 17, 20, 6, 21, 139 samples, respectively. 3,312 genes are retained after excluding genes whose standard deviation are smaller than 50 expression units.

AMLALL data set comprises of 72 samples, in which 25 subjects are acute myeloid leukemia (AML) and 47 subjects are acute lymphoblastic leukemia (ALL). Every sample contain 7,219 gene expression values.

LEU data set comprises of 72 samples of which some are leukemia patients. The data set has 3,571 gene descriptors.

GLIOMA data set contains 50 samples. These are from four classes, which have 14, 14, 7, 15 samples. Each sample contains 4,434 genes.

ProCancer data set contains 89 samples. Each sample contains 15,154 genes. The data set contains two class normal and prostate cancer.

Synthetic data set comprises of 400 samples with 100 features. It has two classes.

Table 1. Data Sets Description

	# OF CLASS	# OF FEATURES	# OF SAMPLE
LUNG	5	3,312	203
AMLALL	2	7,219	72
LEU	2	3,571	72
GLIOMA	4	4,434	50
PROCANCER	2	15,154	89
SYNTHETIC	2	100	400

4.2 EXPERIMENTAL SETUP

The GRM framework is applied the supervised feature selection algorithms such as on ReliefF (RF), Fisher score (Fscore) and information gain (IG). Fisher score is one of the most simple but effective supervised feature selection method. Features with large between-class distance and small within-class distance are preferred by Fisher score method. ReliefF is another representative filter method. The weight of each feature increases if it differs from nearby instances of the other class more than nearby instances of the same class. Information gain measures the reduction of entropy for each feature.

The selected features are given into the cluster based boosting which uses SVM and KNN as classifiers. For each feature selection method, five-fold cross-validation is used to evaluate the performance. The following evaluation metric to measure the redundancy in selected features:

$$\frac{1}{m(m-1)} \sum_{f_i, f_j \in S, i \neq j} A_{i,j} \tag{6}$$

Where S is the set of selected features, m is the number of selected features, f_i, f_j are the features, A_{ij} is the squared cosine similarity defined in the GRM framework. A_{ij} is positive semidefinite to avoid non-convex problem and to obtain the global optimum solution.

4.3 GRM FRAMEWORK FOR CLUSTER BASED BOOSTING

Different supervised feature selection algorithms are used to select the top k features. Their redundancy estimation, boosting accuracy are discussed below.

4.3.1 REDUNDANCY ESTIMATION:

GRM based ReliefF (GRF), GRM based Fishers Score (GFscore) and GRM based Information Gain (GIG) gives the value of refined score. This is obtained by applying ReliefF, Fishers Score and Information Gain in GRM Framework.

From fig. 3 we can infer that the redundancy of GRM based ReliefF is much higher than ReliefF. It is clearly shown in Lung, AMLALL and LEU dataset that the redundancy of GRM based ReliefF is much lower than ReliefF.

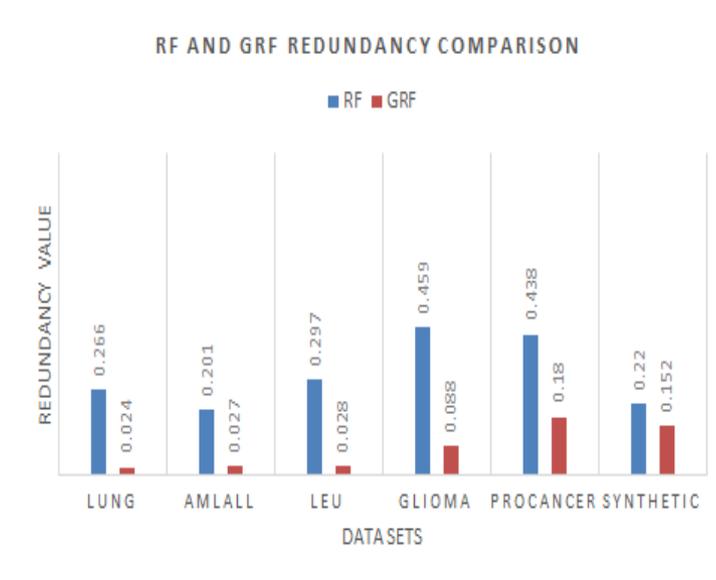


Fig. 3. Redundancy of top 20 features selected by ReliefF and GRM based ReliefF

From fig. 4 we observe that the redundancy of Fishers Score is higher than the GRM based Fishers Score. By comparing with fig. 3 and fig. 5 we observe the redundancy of Fisher Score is higher than other two supervised algorithms. This is clearly shown by Lung and AMLALL data set.

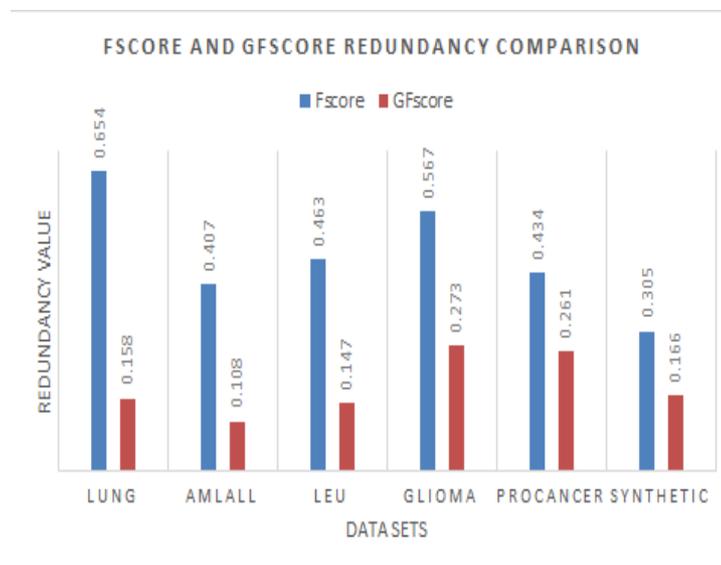


Fig. 4. Redundancy of top 20 features selected by Fishers score and GRM based Fishers score

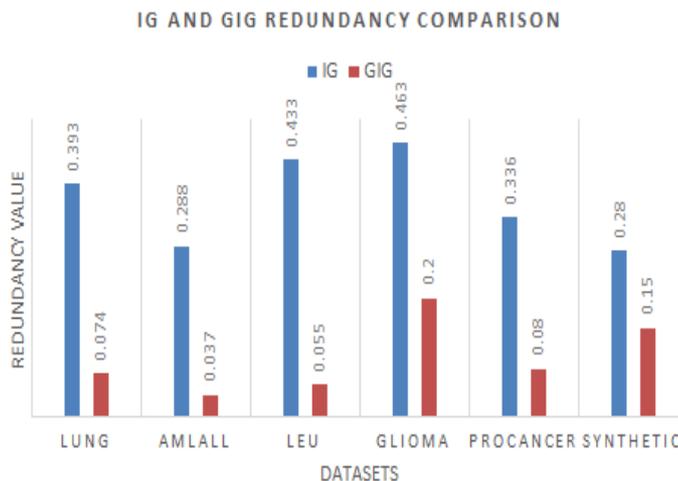


Fig. 5. Redundancy of top 20 features selected by Information Gain and GRM based Information Gain

Fig. 5 records the redundancy of Information Gain and GRM based Information Gain. On all the dataset the redundancy of GIG is much lower than Information Gain.

The reason for the decrease in redundancy in all GRM based supervised algorithm is the features that are globally redundant are expected to get lower refined score z and thus not included in final subset of features.

4.3.2 CLASSIFICATION ACCURACY ESTIMATION

In this study, we compare the performance of top 20 features selected by different supervised feature selection algorithm in terms of accuracy. Support vector machines and K-Nearest Neighbor are the classifiers used.

From table 2 and table 3 we can observe that the features selected by GRM based RF, GRM based IG and GRM based Fishers Score performs better than RF, IG and FScore respectively on all the dataset. Especially on Lung data set the increase in accuracy shows the effectiveness of GRM framework which reduces the redundancy between the selected subset of features and they are expected to be more discriminant.

Table 2. Five cross validation performed on SVM

	RF	GRF	F score	GF score	IG	GIG
LUNG	78.42	89.67	78.42	95.17	78.42	95.64
AMLALL	85.89	88.93	91.61	97.14	93.04	95.54
LEU	94.46	95.21	92.57	93.04	95.71	95.89
GLIOMA	62.00	68.00	60.00	64.00	58.00	64.00
ProCancer	91.05	93.33	93.14	97.78	94.21	96.67
Synthetic	94.75	95.00	94.25	95.50	94.25	95.00

By comparing the table 2 and 3, the top 20 features selected by Information Gain after applying GRM framework performs better than all other supervised feature selection algorithms.

Table 3. Five Cross Validation Performed On KNN

	RF	GRF	F score	GF score	IG	GIG
LUNG	86.81	91.67	84.17	93.35	93.17	94.57
AMLALL	80.54	90.18	83.04	91.61	84.82	94.46
LEU	84.64	94.46	95.89	95.89	91.61	97.14
GLIOMA	56.00	66.00	62.00	94.50	70.00	74.00
ProCancer	86.67	88.76	96.67	98.89	94.44	94.44
Synthetic	92.50	94.00	92.00	94.50	93.00	93.75

4.3.3 BOOSTING ACCURACY ESTIMATION

To show the effectiveness of Cluster Based Boosting technique with GRM feature selection, we compare the performances of adaboost and cluster based boosting technique with top 20 features selected by various supervised feature selection algorithms.

To apply boosting to high dimensional data GRM feature selection is performed. K-Nearest Neighbor and support vector machines is the classifier used in boosting technique. From table 4 we infer that cluster based boosting with features selected from GRM based Fisher Score provides higher accuracy than all other combinations.

By comparing the table 2, 3 with table 4, 5 respectively we infer that the Boosting technique with features selected by GRM framework provides higher accuracy than the supervised learners such as SVM and KNN.

From table 4 and table 5 we observe that the Cluster Based Boosting technique gives higher accuracy than adaboost on all the features selection algorithm.

Table 4. Five Cross Validation Performed On CBB with SVM as Classifier

	GRF-Ada	GRF-CBB	GFS-Ada	GFS- CBB	GIG-Ada	GIG-CBB
LUNG	90.13	91.24	95.5	96.1	95.89	96.20
AMLALL	89.10	90.36	97.14	97.52	95.67	96.23
LEU	95.77	96.14	92.57	93.12	95.71	96.3
GLIOMA	69.45	71.23	66.23	67.56	64.57	67.45
ProCancer	93.25	94.67	97.56	97.81	96.56	97.10
Synthetic	95.10	95.73	95.79	95.89	95.12	95.94

We found that the source of the improved accuracy for CBB with GRM feature selection is by selecting globally non redundant features using GRM framework. Then, the source of CBB is generally the result of predictions on Heterogeneous Struggling and Heterogeneous Prosperous clusters. Taken together, these results support the effectiveness of our CBB selective boosting both in deciding when to boost on the clusters (Heterogeneous Struggling and Heterogeneous Prosperous) to address troublesome areas and when to refrain from boosting (Homogeneous Prosperous) to address label noise.

Table 5. Five Cross Validation Performed On CBB with KNN as Classifier

	GRF-Ada	GRF-CBB	GFS-Ada	GFS- CBB	GIG-Ada	GIG-CBB
LUNG	91.93	92.68	93.56	94.67	94.80	95.56
AMLAL	91.1	92.45	91.65	93.01	94.67	95.6
LEU	95.07	96.14	95.95	96.89	97.24	97.56
GLIOMA	66.45	70.23	65.27	67.56	66.07	68.40
ProCancer	89.10	90.45	98.89	98.89	94.56	95.37
Synthetic	94.57	95.73	94.56	95.89	94.76	95.04

5 CONCLUSION

We propose a cluster based boosting using x-means clustering in high dimensional data. We apply Global Redundancy Minimization framework on high dimensional data to obtain a more compact and non-redundant features. CBB address the various limitations on the supervised learners. We mitigate limitations such as filtering for subsequent functions and overfitting of subsequent functions by using appropriate amount of boosting and by learning only similar data. And it is shown that CBB achieves higher performance than other boosting, we intend to continue our research in several issues such as fine tuning the CBB parameters for LE (δ_1 in (2)) and MLE (δ_2 in (3)) on h dataset. We intend to investigate how to automatically set these parameters based on the data structure and properties.

REFERENCES

- [1] T.G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization," *Journal of Machine Learning*, vol.40, pp138-157, August 2000.
- [2] R. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2012.
- [3] D. Mease and A. Wyner, "Evidence contrary to the statistical view of boosting," *Journal of Machine Learning Research*, vol. 9, pp. 131–156, Feb. 2008.
- [4] P. Long and R. Servedio, "Random classification noise defeats all convex potential boosters," *Journal of Machine Learning*, vol. 78, pp. 287–304, Dec. 2009.
- [5] Vezhnevets and O. Barinova, "Avoiding boosting overfitting by removing confusing samples," *Proceedings of the 18th European Conference on Machine Learning*, pp430-441, 2007.
- [6] L. Dee Miller and Leen-Kiat Soh, "Cluster based boosting," *IEEE transactions on Knowledge and Data engineering*, vol. 27, no. 6, pp.1491-1504, June 2015.
- [7] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics Computation Biology*, vol. 3, no. 02, pp. 185–205, 2005
- [8] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min redundancy," *IEEE Transaction in Pattern Analysis Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
- [9] De Wang, Feiping Nie, "Feature selection via Global Redundancy Minimization, " *IEEE transactions in Data and Knowledge Engineering*, vol. 27, pp. 2743-2755, 2015.
- [10] C. Seiffert, T.Khoshgoftaar, J.Hulse, and A.Naplitano, "Resampling or reweighting: A comparison of boosting implementations," in *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, 2008, pp. 445–451.
- [11] L. Reyzin and R. Schapire, "How boosting the margin can also boost classifier complexity," in *Proceedings of the International Conference on Machine Learning*, pp. 753–760, 2006.
- [12] W. Gao and Z-H. Zhou, "On the doubt about margin explanation of boosting," *Artificial Intelligence*, vol. 203, pp. 1–18, Oct. 2013.
- [13] D.-S. Kim, Y.-M. Baek, and W.-Y. Kim, "Reducing overfitting of adaboost by clustering-based pruning of hard examples," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, pp. 1–3, 2013.
- [14] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the GINI index and information gain criteria," *Univ. Neuchatel, Neuchatel, Switzerland*, 2000.
- [15] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1/2, pp. 273–324, 1997.
- [16] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, pp. 78–87, 2012.
- [17] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA, USA: MIT Press, 2001.
- [18] D. Pelleg and A. Moore, "X-Means: Extending K-means with efficient estimation of the number of clusters," *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 727–734.
- [19] M. B. Priestley, *Spectral Analysis and Time Series*. New York, NY, USA: Academic Press, 1981, pp. 375.
- [20] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 416–431, Mar. 2006.