Efficiently Clustering of High Dimensional Data using Attribute Selection

Miss Palak V. Desai
Department of Computer Engineering
Parul Institute of Engineering and Technology
Vadodara, India
desai.palak27@gmail.com

Mr Arpit Rana
Department of Computer Science & Engineering
Parul Institute of Engineering and Technology
Vadodara, India
to_arpitransa@yahoo.com

***Abstract*— Clustering real word data sets is often hampered by the curse of Dimensionality: many real world data sets consist of high dimensional feature space. Clustering is the technique which groups the similar type of objects into one cluster and dissimilar type of objects into another cluster. Most of the clustering algorithms fail to generate meaningful results because of the inherent sparseness of data space. Attribute selection (feature selection) is a pre-processing step which handles high dimensional data by selecting the subset of attributes from original dataset and removing the irrelevant and redundant attributes and improving quality of clustering. In this paper hybrid approach is used to increase quality of clustering. First applying efficient attribute selection algorithm and then apply K mean algorithm on reduced dimension. This will ultimately generate high quality cluster.**

Keywords—Clustering ,High dimensional data, attribute selection, K Mean

I. INTRODUCTION

With the rapid increase of dimensionality of data such as genomic micro-array data, text categorization and digital images, feature selection has become an important issue, though it is still considered to be an intractable problem in machine learning and data mining. In many applications, the data is usually represented by a huge number of features (attributes), and the raw data often contain many uninformative (irrelevant and redundant) ones which may largely degrade the learning performance and compromise the quality of clustering. High dimension increases the complexity of understanding the dataset itself and applying the algorithm, since many algorithms are sensitive to largeness or high-dimensionality or both [1].

Therefore, to retain important features and remove irrelevant and redundant ones, various robust and effective feature selection algorithms have been introduced recently [2].There are three major benefits of feature selection (FS): (1) improves the prediction performance of the predictors; (2) helps predictors do faster and more cost-effective prediction; and (3) provides a better understanding of the underlying process that generated data.

Methods in unsupervised feature selection also can be classified into two categories: filter category and wrapper category. Methods in filter category consider selection of features based on intrinsic properties of the data without involving any clustering algorithms. The principle is that any feature carrying little or no additional information beyond that subsumed by the remaining features, is redundant and should be removed. On the other hand, the wrapper approaches wrap feature selection process around a specific clustering algorithm for maximizing clustering performance. Wrapper category involves searching through various feature subsets, followed by the evaluation of each of them using evaluation criterion [3].

In this paper first we apply feature selection algorithm which produced important reduced features. Apply clustering algorithm on the resultant data of first stage. Some clustering algorithms like k means, k-medoid, DBSCAN, Hierarchical clustering algorithms, CLARA which do not support high dimensional dataset. Thus applying the feature selection as a pre processing step for the clustering and make it possible to handle the high dimensional dataset.

II. FEATURE SELECTION ALGORITHM

In this section I will give the summary of different algorithm and the different approaches used by different authors.

A. Correlation-Based Filter Approach

This involves two aspects: (1) how to decide whether a feature is *relevant* to the class or not; and (2) how to decide whether such a relevant feature is

redundant or not when considering it with other relevant features.
The answer to the first question can be using a user defined threshold *SU* value.

$$SU(X, Y) = 2\left[\frac{IG\left(\frac{X}{Y}\right)}{H(X) + H(Y)}\right] \quad (1)$$

The answer to the second question is more complicated because it may involve analysis of pair wise correlations between all features (named *F*-correlation). To solve this problem, we propose our method below. Since *F*-correlations are also captured by *SU* values, in order to decide whether a relevant feature is redundant or not, we need to find a reasonable way to decide the threshold level for *F*-correlations as well.
FCBF [4] algorithm is as follow.

1) Data set with *N* features and a class.

2) The algorithm finds a set of predominant features *Sbest* for the class concept. It consists of two major parts. In the first part, it calculates the *SU* value for each feature, selects relevant features into *S'list* based on the predefined threshold δ, and orders them in descending order according to their *SU* values. In the second part it further processes the ordered list *S'list* to remove redundant features and only keeps predominant ones among all the selected relevant features.

3) Feature *Fp* that has already been determined to be a predominant feature can always be used to filter out other features that are ranked lower than *Fp* and have *Fp* as one of its redundant peers.

4) The iteration starts from the first element in *S'list* and continues as follows.

5) For all the remaining features (from the one right next to *Fp* to the last one in *S'list*), if *Fp* happens to be a redundant peer to a feature *Fq*, *Fq* will be removed from *S'list*.

6) After one round of filtering features based on *Fp*, the algorithm will take the currently remaining feature right next to *Fp* as the new reference to repeat the filtering process.

7) The algorithm stops until there is no more feature to be removed from *S'list*.

B. *Clustering Ensemble for Unsupervised Feature Selection*

Algorithm selects the feature subset through combining the clustering ensembles method and the modified RReliefF algorithm. In particular, the task of feature selection involves two steps. The method firstly obtains multiple clustering from K means algorithms in different randomly selected feature subspace and generates a single consensus co-association matrix. Then the modified RReliefF algorithm ranks all features depending on the element in co-association matrix.

The pseudo code of the CEFS [3] algorithm is described as follows:

Input: The number of cluster *p*, dataset *D*, the number of clustering solution in ensemble *T*, the *i*-th feature *i F* , the number of iterations *k*, the number of nearest neighbours *m*.

Output: $W[F_i]$, $i=1 \ldots M$, the weight of feature *Fi*

1) For l=1 to *T* Do
   S_l =randomly select a half of full features;
   $\lambda^{(l)}$ =k-means (S_l ,*p*); //generate Clustering in selected feature subspace and *p* maybe either same or different in individual component ;
   Generate co-matrix *Cl* ;
   Endfor
2) *C* =Combine $\{C^{(1)}, C^{(2)}, \ldots, C^{(T)}\}$;
3) Set every weight $W[F_i] = 0.0$, $P_{diffc} = 0.0$, $P_{diffFi} = 0.0$, $P_{diffc\&diffFi} = 0.0$;
4) For i=1 to *k* Do
   Randomly select an instance *i* from *D*;
   Find *m* of its nearest neighbours;
   For each neighbour update P_{diffc} ;
   For each neighbour and each attribute update P_{diffFi} and $P_{diffc\&diffFi}$
   Endfor
5) For each attribute get $W[F_i]$ using (5);
6) Rank all features depending on $W[F_i]$.

P_{diffc} = *P(difference prediction/nearest instances)* *(2)*

P_{diffFi} =*P(difference value of* F_i */nearest instances)* *(3)*

$P_{diffc\&diffFi}$ = *P(difference prediction and diff. value of* F_i */nearest instances)* *(4)*

W [F_i] can be calculated by following equation

$$P_{diffc\&diffFi} / P_{diffc} - (P_{diffFi} - P_{diffc\&diffFi}) / (k - P_{diffc}) \quad (5)$$

The clustering ensembles method can achieve a better clustering solution if the number of clusters of individual component algorithm is set larger than the real number of clusters.

C. *Text Clustering with Feature Selection (**TCFS**) by Using Statistical Data*

This method is used in the text categorisation process and provides the accurate and efficient cluster.

- New feature selection method CHIR.

This method is based on the $\chi 2$ statistic. This method can measure the positive term category dependency. This method is select the terms which are relevant in the categories and remove the irrelevant and redundant term [5].

TCFS algorithm

1) Use clustering algorithm like K-means on dataset and get the initial cluster.

2) Perform supervised feature selection method such as CHIR, on the dataset by using current clustering result as the call label information of the document.

3) Untouched the selected term but weight on the unselected feature is reduced by predetermine factor.

4) Calculate the k centroids in new feature space.

5) For each document in the corpus, compute the clustering criterion function with each cluster centroids in the new feature space.

6) Repeat 2 to 5 until convergence.

Here performing the feature selection method based on the information cluster obtained during the clustering process which improves the performance of the clustering accuracy.

D. *Novel Unsupervised Feature Selection Method for Bioinformatics Data Sets through Feature Clustering*

This Feature Selection through Feature Clustering (FSFC) algorithm groups the features into different clusters based on the similarity therefore the similar features are in same cluster. This feature selection method consists of two major steps: first the entire feature set is partitioned into different homogeneous subsets (clusters) based on the feature similarity, and then a representative feature is selected from each cluster. Such representatives (features) constitute the optimal feature subset [6].

The FSFC algorithm is summarized as below:

1) Select the dataset which contain the number of features.

2) Calculate the MICI for each cluster.

MICI is calculated using

$$\text{var}(x) + \text{var}(y) - \sqrt{(\text{var}(x) - \text{var}(y))^2 + 4\text{cov}(x,y)^2} \quad (6)$$

Where var = variance,

cov = covariance between two Variables x, y.

3) Select the minimum clusters and merge that cluster and feature information the single cluster.

4) Select the top k cluster in the hierarchical cluster tree.

This algorithm does not require the feature search. This algorithm is used for the high dimensional dataset. FSFC does not require the class label information for the supervised and unsupervised learning process.

III. K-MEANS ALGORITHM

K-means is a prototype-based, simple partition clustering technique which attempts to find a user-specified k number of clusters. These clusters are represented by their centroids. A cluster centroid is typically the mean of the points in the cluster. This algorithm is simple to implement and run, relatively fast, easy to adapt, and common in practice.

This algorithm consist of two separate phases: the first phase is to select k centroids randomly, where the value of k is assigned earlier. The next phase is to assign each data object to the nearest centre. Euclidean distance method is generally used to determine the distance between each data object and the cluster centres. When all the data objects are assigned to each of the clusters, the cluster centres recalculation is done. This iterative process continues repeatedly until the criterion function of finding new cluster centres becomes minimum or reduced [7].

1) Randomly select k data object from dataset D as initial cluster centres.
2) Repeat
   Calculate the distance between each data object di (1<=i<=n) and all k cluster centres' cj(1<=j<=n) and assign data object di to the nearest cluster.
3) For each cluster j(1<=j<=k), recalculate the cluster centres.
4) Until no change in the cluster centre.

Figure 1. K-means algorithm

IV. CONCLUSION

This paper presents various aspects of feature selection. This paper has introduced several scores for computing the feature importance and clustering based framework of feature selection. For clustering high dimensional data which suffer from curse of dimensionality, we first apply one of the efficient feature selection algorithms and reduce irrelevant and redundant features. As a result we can get most relevant features and on these features we apply k-means algorithm for the purpose of clustering. By this way we can increase the quality of clustering having High dimensional data.

V. REFERENCES

[1] Naijun Wu, Xiuyun Li, Jie Yang, Peng Liu, "Improved Clustering Approach based on Fuzzy Feature Selection", IEEE, 2007.

[2] Shengyi Jiang, Lianxi Wang," Unsupervised Feature Selection Based on Clustering", IEEE, pp. 263, 2010.

[3] Yihui Luo, Shuchu Xiong," Clustering Ensemble for Unsupervised Feature Selection", Sixth International Conference on Fuzzy Systems and Knowledge Discovery, pp.445, 2009.

[4] Lei Yu, Huan Liu," Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, 2003.

[5] Yanjun Li, Congnan Luo and Soon M. Chung, Member, IEEE," Text Clustering with Feature Selection by Using Statistical Data," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. XX, NO. YY, 2008.

[6] Guangrong Li , Xiaohua Hu , Xiajiong Shen , Xin Chen , Zhoujun Li "A Novel Unsupervised Feature Selection Method for Bioinformatics Data Sets through Feature Clustering", Granular Computing, 2008. GrC 2008. IEEE International Conference on 26-28 Aug. 2008.

[7] H.S. Beheraa, Abhishek Ghosh, Sipak Ku.Mishra," A New Improved Hybridized K-MEANS Clustering Algorithm with Improved PCA Optimized with PSO for High Dimensional Data Set", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012.