OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Design an Efficient Clustering Technique Based On Feature Subset Selection in High Dimensional Data

Ashok Kumar B.[1], Devan D. P.[2]

Abstract: *Feature range involves identifying a subset of the most useful characteristic that produces compatible results as the original entire set of Characteristic. A feature range algorithm may be calculated from both precision and proficient point of view. While the precision is for find the subset of characteristics, the proficient is related to the time requirement of subset. The algorithm works in three steps in first step characteristics are spited into clusters in second step assign the centroids in the each clusters, in third step strongly related to target classes is selected from each cluster to from subset characteristics. These steps able to perform the precision and proficient process in the cluster based subset. Using the minimum spanning tree clustering method in precision and comparing the cluster is able to perform high in the proficient. The goal of our method is to perform better cluster discovery and able to reduce the computation time in the clustering.*

Keywords: *Clustering, Irrelevant, precision, proficient, reduntent.*

I. Introduction

The mean of selecting a subset of good characters with respect to the target concepts, feature subset range is an precision way for reducing dimensionality, removing unrelated data, increasing knowledge accuracy, and improving result unambiguousness. Many feature subset range methods have been proposed and considered for machine knowledge applications. They can be divided into some category: the rooted, covering, clean, and cross approaches.

The rooted method incorporate feature range as a part of the training process and are usually specific to given knowledge algorithms, and therefore may be more proficient than the other three category. conventional machine knowledge algorithms like decision trees or artificial neural networks are examples of rooted approaches. The covering method use the predictive precision of a prearranged learning algorithm to determine the goodness of the chosen subsets, the precision of the learning algorithms is usually high. However, the generality of the choosen features is limited and the computational complexity is large. The clean method are independent of learning algorithms, with good simplification. Their computational complication is small, but the precision of the knowledge algorithms is not guaranteed. The cross method are a grouping of clean and covering methods, by using a clean method to shrink search gap that will be measured by the subsequent covering. They mainly focus on combining clean and covering method to achieve the best possible performance with a particular learning algorithm with similar time complexity of the clean methods. The covering methods are computationally expensive and tend to over fit on training sets. The clean method, in addition to their simplification, are usually a high-quality preference when the number of features is very large. Thus, we will focus on the clean method in this paper.

With respect to the clean feature selection method, the application of cluster analysis has been established to be more useful than traditional feature selection algorithms.

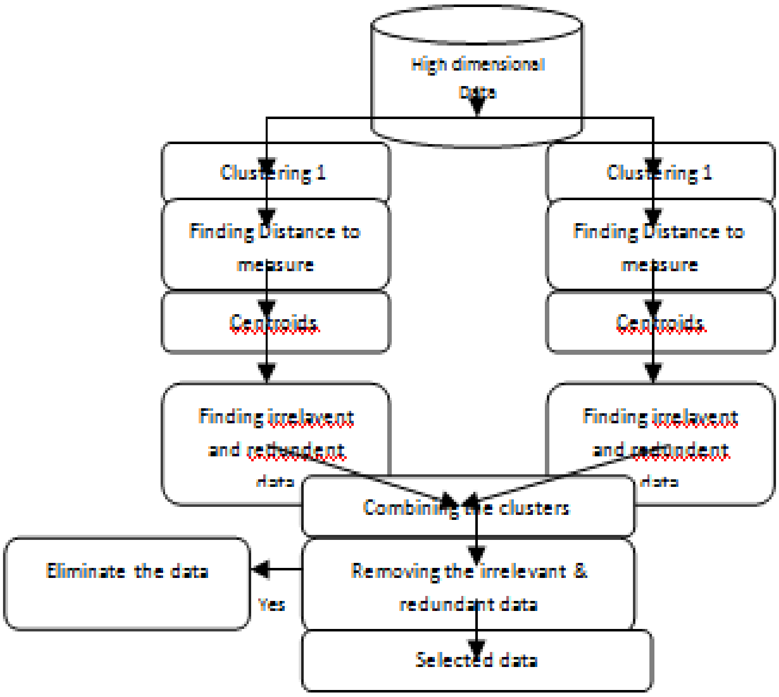
In cluster analysis, partition methods have been well studied and used in many applications. Their outcome have, at times the best concurrence with human performance. The general partition clustering is simple. In our study, we apply partition clustering methods to features. In exacting, we accept the minimum spanning tree (MST) based clustering algorithms, because they do not suppose that data points are grouped approximately centers or alienated by a normal characteristics and have been widely used in practice.

Based on the MST method, we propose a new algorithm. The algorithm works in three steps in first step characteristics are spited into clusters in second step assign the centroids in the Each clusters, in third step strongly related to target classes is selected from each cluster to from subset characteristics. Features in different clusters are relatively independent; the clustering-based strategy of new algorithm has a high probability of producing a subset of useful and independent features. The proposed new algorithm was tested in text data sets. The investigational results show that, compared with feature subset range algorithms, the proposed algorithm not only reduces the quantity of facial appearance, but also improves the presentation of the four well-known different types of classifiers.

In this paper we propose the precision and proficient by using the algorithms and the methods. The partition algorithm is for splitting the clusters in the range after that able to assign the centroids. These centroids are assign by the distance or by the randomly. If we assign distance measure is using the some mathematical formulation to find the centroids. After finish finding the centroids comparing the centroids wheather have a duplicate copy or a noisy data is find in it just remove from the dataset otherwise select and shortlist it. Finally compare the data with the other using the Sequesser algorithm. In this algorithm it eliminate the irrelevant and redundant data in the data set. Finally the result shows that accuracy and efficiency of the dataset.

II. Methodology

ARCHITECTURE


High dimensional Data
Clustering 1
Finding Distance to measure
Centroids
Clustering 1
Finding Distance to measure
Centroids
Finding irrelavent and rethundent data
Finding irrelavent and redundent data
Combining the clusters
Eliminate the data
Yes
Removing the irrelevant & redundant data
Selected data

In this paper using some methods they are
i)**Splitting data into Clusters**
ii)**Assigning Centroids**
iii)**Eliminate the Irrelevant and redundant data**

i)Splitting data into clusters

The high dimensional data are spitted into clusters. These clusters are split data based on the types of the file or related to that file. In cluster the files can be repeated or may be a irrelevant file in it. So the comparison of the file take more time in the cluster for that reason using the interclustring and intraclustering in the cluster comparison.

Interclustering – Comparing with other clusters.
Intraclustering- Comparing within the clustering.

Interclustering compare the files with other cluster based on the related data in the cluster. The intraclustering compare the files in the same cluster to check having any duplicate file are located in the cluster. These two technique are able to reduce some of the processing time in the comparison. The files are spitted based on related data (i.e) it may be a image files , document files, rar files, etc… some of the files may split by the size also. The cluster evaluation measure based on distance does not identify a feature subset that allows the classifiers to improve their original performance accuracy. Further more, even compared with other feature selection methods.

The dataset are partitioned into some of the clusters based on the partition algorithm. The dataset split your files in 3 ways.

1.Splitting by distance file
2.classification
3.classification also using a fasta file.

1.Splitting by distance file

For the distance file method, you need only provide your distance file and mothur will split the file into distinct groups.

2.Classification

For the classification method, to provide distance file, taxonomy file, and set the splitting method to classify. After that set the tax level to split by mothur. It split the sequences into distinct taxonomy groups, and split the distance file based on those groups.

3.Classification also using a fasta file

For the classification method using a fasta file, to provide fasta file, names file and taxonomy file. Set the tax level to split by mothur and split the sequence into distinct taxonomy groups, create distance files for each grouping.

The split method parameter allows to specify how to split files before the cluster, default=distance, options distance, classify or fasta .

Algorithm 1

<u>Partition Algorithm</u>

```
Function partition (S)
n □ |S|
N □ sum(S)
P □ emptyBoolean table of size ((N/2) +1) by (n +1)
Initialize top row (P(0,x)) of P to True
Initialize leftmost column (P(x,0)) of P, execute for P (0,0) to false
For I from 1 to (n/2)
For j from 1 to n
P(i, j) □ P(i,j-1) or P(i-S[j-1], j-1)
Return P((n/2),n)
```

The above algorithm 1 is split the data into number of clusters based on the size.

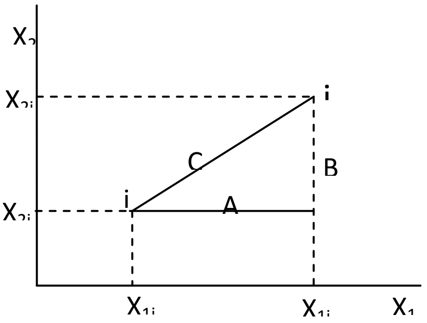
ii)**Assigning Centroids**

In this part the each clusters generate a centroids . This centroids are measure by distance basis or selected as randomly. In distance basis the data are check all near by data to assign a centroid. In randomly basis it just assign any one data as centroid. The centroids are able to compare the data in the cluster and with the other cluster. It able to reduce the replica data in the dataset.

- Distance based
- Selecting Randomly

Distance based:

This method is used to find the distance between each data in the dataset by using the mathematical calculation as shown in the below



$$D(i,j)=\sqrt{A^2+B^2} = \sqrt{(X_{1i}-X_{1j})^2 + (X_{2i} - X_{2j})^2}$$

The Above equation is used to find the distance between the data present in the dataset. This type is able to calculate the equation of each data and form the centroid. All data is calculated by the equation and then it assign the centroid.

Selecting Randomly

This method able to select the centroid by assigning the centroid randomly. This makes easy to find centroid but difficult to compare the data.

iii) Eliminate the Irrelevant and redundant data

The irrelevant feature removal is straightforward once the right relevance measure is defined or selected, while the redundant feature elimination is a fragment of sophisticated. In this module able to reduce the same copy of data in the cluster and remove the noise data also. The processing is based on the mathematical format. The similarity between two documents must be measured in some way if a clustering algorithm is to be used. There are a number of possible measures for computing the similarity between documents, but the most common one is the cosine measure, which is defined as

$$cosine(\,\boldsymbol{d}1, \boldsymbol{d}2\,) = (\boldsymbol{d}1 \tilde{.}\, \boldsymbol{d}2) \,/\, \|\boldsymbol{d}1\|\;\|\boldsymbol{d}2\| \quad ....(1)$$

where $\tilde{.}$ indicates the vector dot product

$\|\boldsymbol{d}\|$ - is the length of vector ***d***.

d - Vector

Given a set, *S*, of documents and their corresponding vector representations, we define the ***centroid*** vector ***c*** to be

$$C=\frac{1}{|S|}\sum_{d\in S} d \qquad ..(2)$$

Where C - Centroid
S – Set
d - Vector

which is nothing more than the vector obtained by averaging the weights of the various terms present in the documents of *S*. Analogously to documents, the similarity between two centroid vectors and between a document and a centroid vector are computed using the cosine measure,

$$cosine(\,\boldsymbol{d}, \boldsymbol{c}\,) = (\boldsymbol{d} \tilde{.}\, \boldsymbol{c}) \,/\, \|\boldsymbol{d}\|\;\|\boldsymbol{c}\| = (\boldsymbol{d} \tilde{.}\, \boldsymbol{c}) \,/\, \|\boldsymbol{c}\| \quad ...(3)$$

$$cosine(\,\boldsymbol{c1}, \boldsymbol{c2}) = (\boldsymbol{c}1 \tilde{.}\, \boldsymbol{c}2) \,/\, \|\boldsymbol{c1}\|\;\|\boldsymbol{c2}\| \quad ...(4)$$

Where d - Vector
C - Centroid
C1 - Centroid1
C2 - Centroid 2

The cosine measure is used to compute which document centroid is closest to a given document. While a median is sometimes used as the centroid for clustering, we follow the common practice of using the mean. The mean is easier to calculate than the median and has a number of nice mathematical properties. For example, calculating the dot product between a document and a cluster centroid is equivalent to calculating the average similarity between that document and all the documents that comprise the cluster the centroid represents. This observation is the basis of the “intracluster similarity” Mathematically. compare the same data and noise data in the dataset. This make the comparison easy (i.e) the centroid comparison is make less time conception to give a precision and provesion data in the dataset. describe the comparing of clusters i.e. comparing of one cluster with the other clusters if its having the same dataset it destroy the any one cluster which are same.

Algorithm 2

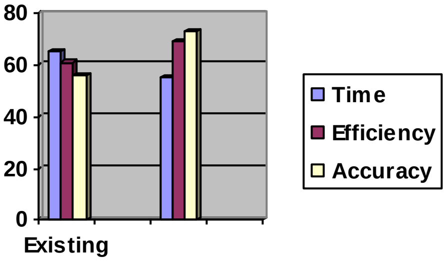
<u>SQUEEZER ALGORITHM</u>

```
Squeezer(D,s)
Begin
While(D has unread tuple)
{
Tuple= getCurrentTuple(D)
If(tuple.tid==1)
{
Addnewclusterstruture(tuplie.tid)
}
Else
{
For each existed cluster C
simComputation(C,tuple)
get the max value of similarity : sim_max
get the corresponding cluster Index:Index
if sim_max>=s
addtupletocluster(tuple, index)
else
addnewclusterstructure(tuple,tid)
}
}
Outputclusteringresult()
End
```

The above algorithm is used to eliminate the irrelavent and redundant data in the dataset and able to reduce the time. This algorithm is uses in the each cluster so the elimination the replica is easy in the cluster.

Experimental Result

In the existing system the FAST algorithm accuracy is very low compare to the proposed system. The proposed system compare with the centroid is able to decrease the time and increase the accuracy in dataset. The dataset are spited and then compareing but in the existing system it compare hole dataset it take more time to comopare.



III. Conclusion

The proposed work have presented a novel clustering-based feature subset selection algorithm for high dimensional data. The algorithm involves (i) partitioning the minimum spanning tree and selecting representative features, (ii) constructing a minimum spanning tree from relative ones, and (iii.) removing irrelevant and redundant features A cluster consists of features. Each cluster is treated as a single feature and thus dimensionality is drastically reduced. The experimental result of proposed system method is to perform better cluster discovery and able to reduce the computation time in the clustering.

In future Irrelevant and redundant is to be determined from the Subset selection and clustering. Then combining the process are to be apply to the Selected clusters for mining the clustering data.

REFERENCE

[1] Qinbao Song, Jingjie Ni and Guangtao Wang, "A Fast Clustering Based Feature Subset Selection Algorithm For High Dimentional Data" IEEE Transactions on Knowledge and Data Engineering Vol:25 No:1 Year 2013.

[2] Tajunisha N, saravanan V "New Approach To Improve The Clustering Accuracy using Information Genes For Unsupervised Microarray Datasets," International Journal of Advanced Science and Technology Vol. 27, February, 2011.

[3] Samir Brahim Belhaouari "Fast and Accuracy Control Chart Pattern Recognition using a New cluster-k-Nearest Neighbor" World Academy of Science, Engineering and Technology 25 2009.

[4] H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features," Artificial Intelligence, vol. 69, nos. 1/2, pp. 279-305, 1994.

[5] A. Arauzo-Azofra, J.M. Benitez, and J.L. Castro, "A Feature Set Measure Based on Relief," Proc. Fifth Int'l Conf. Recent Advances in Soft Computing, pp. 104-109, 2004.

[6] L.D. Baker and A.K. McCallum, "Distributional Clustering of Words for Text Classification," Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in information Retrieval, pp. 96-103, 1998.

[7] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," IEEE Trans. Neural Networks, vol. 5, no. 4, pp. 537-550, July 1994.

[8] D.A. Bell and H. Wang, "A Formalism for Relevance and Its Application in Feature Subset Selection," Machine Learning, vol. 41, no. 2, pp. 175-195, 2000.

[9] J. Biesiada and W. Duch, "Features Election for High-Dimensional data a Pearson Redundancy Based Filter," AdvancesinSoftComputing,vol.45,pp.242-249,2008.