IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org
Volume 2, Issue 11 (November2012), PP 01-07

Data Clustering With Leaders and Subleaders Algorithm

Srinivasulu M[1],Kotilingswara Rao R[2], Baji Mohammed[3] , R V S P Kumar[4]

[1,2,3,4] (2/2 M.Tech, Computer Science and Systems Engineering Dept, AUCE (A), Visakhapatnam, AP, India)

Abstract: *In this paper, an efficient hierarchical clustering algorithm, suitable for large data sets is proposed for effective clustering and prototype selection for pattern classification. It is another simple and efficient technique which uses incremental clustering principles to generate a hierarchical structure for finding the subgroups/subclusters within each cluster. As an example, a two level clustering algorithm—Leaders–Subleaders, an extension of the leader algorithm is presented. Classification accuracy (CA) obtained using the representatives generated by the Leaders–Subleaders method is found to be better than that of using leaders as representatives. Even if more number of prototypes are generated, classification time is less as only a part of the hierarchical structure is searched.*

Keyword: *Data Warehouse ,Data mining, Algorithms(BIRCH,DBSCAN,CURE).*

I. Introduction

1.1 Data warehouse:

A data warehouse is a collection of data from multiple sources, integrated into a common repository and extended by summary information (such as aggregate views) for the purpose of analysis. When speaking of a data warehousing environment, we do not anticipate any special architecture but we address an environment with the following two characteristics:
(1) Derived information is present for the purpose of analysis.
(2) The environment is dynamic, i.e. many updates occur.

1.2 Data Mining

Data mining has been defined as the application of data analysis and discovery algorithms that under acceptable computational efficiency limitations produce a particular enumeration of patterns over the data. e.g., clustering, classification and summarization. Typical results of data mining are as follows:

- Clusters of items which are typically bought together by some set of customers (clustering in a data warehouse storing sales transactions).Symptoms distinguishing disease A from disease B (classification in a medical data warehouse).Description of the typical WWW access patterns (summarization in the data warehouse of an internet provider).

1.3 Data Mining Essentials:

Data Mining refers to extracting or mining knowledge from large amounts of data. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery. The amount of data kept in computer files and databases is growing at a phenomenal rate. At the same time, the users of these data are expecting more sophisticated information from the data set. For example a marketing manager is no longer satisfied with a simple listing of marketing contacts, but wants detailed information about customers past purchases as well as predictions of future purchases. For this type of data processing simple structured/query language are not adequate. So to meet these needs or to support these increased demands for information we need some well-sophisticated efficient data processing techniques.
Data Mining algorithms can be characterized as consisting 3 parts:
Model: The purpose of the algorithm is to fit a model to the data.
Preference: Some criteria must be used to fit one model over another.
Search: All algorithms require some techniques to search the data.

1.4 Cluster Analysis:

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group in many applications Cluster analysis has been widely used in numerous applications including Pattern

Recognition, Data Analysis, Image Processing and Market research etc. By clustering one can identify dense and sparse regions and therefore discover overall distribution patterns and interesting correlations among data attributes. As a data mining function, cluster analysis can be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster and to focus on a particular set of clusters for further analysis.

1.5 Requirements of Clustering:

In Data Mining, efforts have focused on finding methods for efficient and effective cluster analysis in larger databases. Active themes of research focus on

- The Scalability of clustering methods, The high-dimensional clustering techniques. The effectiveness of the methods for clustering complex shapes and types of data. The methods for clustering the mixed, numerical and categorical data in large databases.

1.6 Major Clustering Methods:

Before illustrating the different categories of clustering methods, summarize the basic features of clusters and cluster problem. Basic features of Clustering (as opposed Classification).

- The number of clusters is not known. There may not be any prior knowledge concerning the clusters.
- Cluster results are dynamic.

The Clustering problem is stated as:

Given a database D={t1,t2,….tn} of tuples and an integer value K,the clustering problem is to define a mapping f: D ->{1,….,k} where each t_i is assigned to one cluster K_j , 1≤ j ≤ K.A cluster K_j , contains precisely those tuples mapped to it, i.e,K_j={t_i /f(t_i)=K_j ,1 ≤ i ≤ n and t_i ε D}.

Here the number of clusters is an input value k. The actual content of each cluster K_j ,1 ≤ j ≤ k,is determined as a result of the function definition. Without loss of generality, the result of solving a clustering problem is that a set of clusters is created: K={K_1,K_2,K_3,…,K_k}.

There exit a large number of clustering algorithms. The choice of clustering algorithm depends both on the type of data available and on the particular purpose and application. If cluster analysis is used as a descriptive or exploratory tool, it is possible to try several algorithms on the same data.

Partitional Methods:

With Partitional Clustering, the algorithm creates only one set of clusters. These approaches use the desired number of clusters to drive how the final set is created. a partitioning method construct K partitions of the data, where each partition represents a cluster and k ≤ n. That is, it classifies the data into k groups, which together satisfy the following requirements:

- Each group must contain atleast one object. Each object must belong to exactly one group/cluster.

Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. To achieve global optimality in partitioning based clustering would require the exhaustive enumeration of all of the possible partitions and The most popular heuristic methods are:

- K-Means
- K-Medoids/PAM

1.7 Clustering Large Databases:

When clustering is used with dynamic databases, the classic clustering algorithms may not be appropriate. Firstly, they all assume that sufficient main memory exist to hold the data to be clustered and the data structures needed to support then. Performing I/Os continuously through the multiple iterations of an algorithm is too expensive. Because of these main memory restrictions, the algorithms do not scale up to large databases. Clustering techniques should be able to adapt as the database changes. A clustering algorithm performed on large databases should satisfy the following:

1. Require no more than one scan of the database. Have the ability to provide the status and “best answer so far during the algorithm execution.
2. Be suspendable, stoppable and resumable. Be able to update the results incrementally as data are added or removed from the database. Work with limited main memory. Be capable of performing different techniques for scanning the database.
3. This may include sampling. Process each tuple only once.

Algorithms used for clustering large databases are:

- BIRCH
- DBSCAN

- CURE

II. Equations And Algorithm

2.1 Proposed method

Leader is an incremental algorithm in which L leaders (Lds) each representing a cluster are generated using a suitable threshold value. As an extension of leader algorithm, we have implemented Leaders–Subleaders algorithm. In this method, after finding L leaders using the leader algorithm, subleaders (Sublds) are generated within each cluster represented by a leader, choosing a suitable subthreshold value. Thus, Leaders–Subleaders (Lds-Sublds) algorithm creates L clusters with L leaders and SLi subleaders in the ith cluster as shown in Fig. 1.

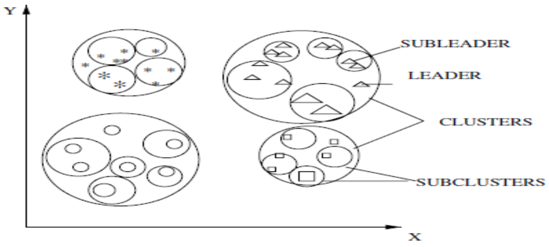


Fig. 1. Clusters in Leaders–Subleaders algorithm.

Subleaders are the representatives of the subclusters and they in turn help in classifying the given new/test pattern more accurately. This algorithm can be used to generate a hierarchical structure as shown in Fig. 2 and this procedure may be extended to more than two levels. An h level hierarchical structure can be generated in only h database scans and is computationally less expensive compared to other hierarchical clustering algorithms. Number of database scans is 6h since the number of training patterns to be scanned during clustering decreases as h increases and also we can efficiently manage the memory.

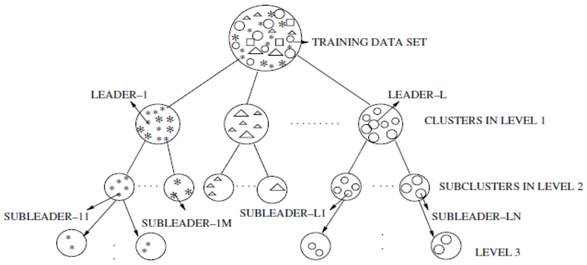


Fig.2. Hierarchical structure generated using the proposed algorithm

Leaders–Subleaders algorithm requires only two database scans and can be used to find the subgroups/subclusters as required in certain applications. In optical character recognition data set, there are 10 classes corresponding to the digits 0–9. There would be several subgroups/subclusters in each class depending on the attribute values of the features. For example, digits may be printed in different fonts as shown in Fig. 3. They can be categorized into different subgroups. It is necessary to find these subgroups/subclusters also. If a representative from each subgroup is chosen then naturally CA would be improved (Vijaya et al., 2003a). Euclidean distance is used for characterizing dissimilarity between two patterns in case of numerical data set. Euclidean distance between two, d dimensional patterns x and y is given by,

$$\mathbf{Eucl} - \mathbf{dist}(x, y) = \sqrt{\left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)}$$

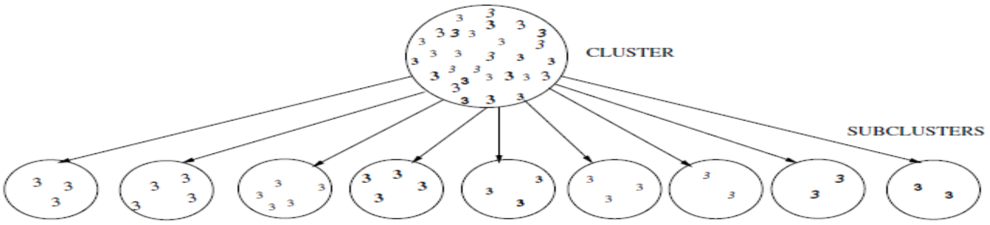


Fig. 3. An example of subgroups/subclusters in printed digits.

Threshold and subthreshold values can be initially chosen depending on the maximum and the minimum euclidean distance values between the objects of a class in case of supervised learning. For unsupervised clustering technique, threshold value should be chosen properly depending on the number of clusters to be generated. If the threshold value is too small then a large number of clusters are generated and if the threshold value is too large then very few clusters are generated. Sub threshold value should be smaller than the threshold value to group the objects of a subgroup / subcluster. Prototypes (representatives of the clusters and subclusters) are generated using the training data set. During classification/testing phase, for every test pattern of the testing data set, the nearest leader is found first and then the nearest subleader in that cluster is determined. The quality of the prototypes is evaluated using the CA obtained for the testing data set (Ananthanarayana et al., 2001). Leaders Subleaders algorithm require two database scans (h ¼ 2) and its time complexity is O(ndh) and is computationally less expensive compared to most of the other partitional and hierarchical clustering algorithms. Space complexity of Leaders–Subleaders algorithm is O((L+SL)d), where L and SL are the total number of leaders and subleaders respectively. For a h level hierarchical structure, the space complexity is $O((\sum^{h}_{j=1} (Lj))d)$, and the sum of the prototypes is less than the total number of patterns—n. The space requirement will be reduced as only these representatives are to be stored in the main memory during the testing phase. Even if more number of prototypes are generated, classification time is less as only part of the hierarchical structure is searched during the testing phase. The algorithms for leader and Leaders–Subleaders are given below.

2.1 Leaders algorithm

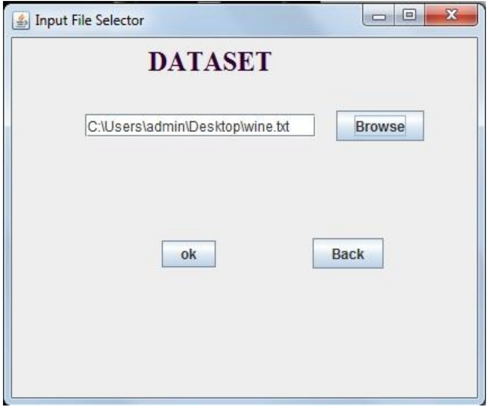
```
Train_leader()
1. Select threshold value
2. Initialize a leader, add it to leader list and set leader counter, L = 1
3. Do for all patterns, i = 2 to n
{
    Calculate the distance with all leaders
    Find the nearest leader
    If (distance with nearest leader < threshold)
    {
        Assign it to the nearest leader
        Mark the cluster number
        Add it to member list of this cluster
        Increment member count of this cluster
    }
    else
    {
        Add it to leader list
        ncrement leader counter, L = L + 1
    }
}
```

2.2 Leaders–Subleaders algorithm:

```
Train_Lds_Sublds()
1. Call Train_leader to generate L clusters/leaders
2. Select subthreshold value (< threshold value)
3. Do for i ¼ 1 to L clusters/leaders
{
    a. Initialize a subleader, add it to subleader list and set counter, SLi = 1
    b. Do for j = 2 to member count of ith cluster
    {
        Calculate the distance with all subleaders
        Find the nearest subleader
        If (distance with nearest subleader < subthreshold)
        {
            Assign it to the nearest subleader
            Mark the subcluster number
            Add it to member list of this subcluster
            Increment member count of this subcluster
        }
else
{
Add it to subleader list
Increment subleader counter, SLi = SLi + 1
}
}
}
4. Initialize counter SL = 0, Do for i = 1 to L{ SL = SL + SLi }
```

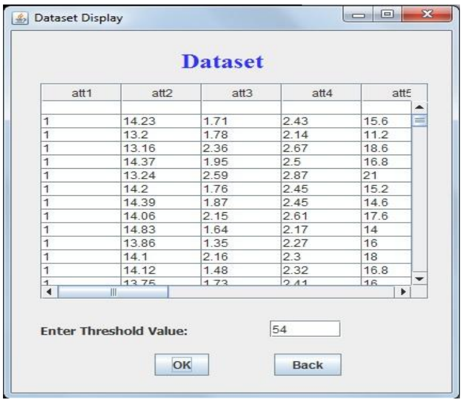
III. Screen Shots And Results

a) Input Screen for wine dataset



C:\Users\admin\Desktop\wine.txt

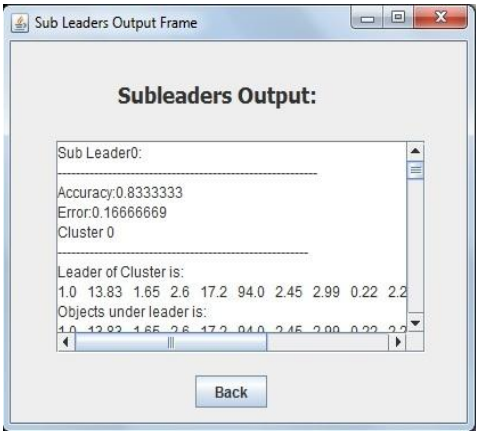
b) Dataset Display Screen for wine dataset



att1	att2	att3	att4	att5
1	14.23	1.71	2.43	15.6
1	13.2	1.78	2.14	11.2
1	13.16	2.36	2.67	18.6
1	14.37	1.95	2.5	16.8
1	13.24	2.59	2.87	21
1	14.2	1.76	2.45	15.2
1	14.39	1.87	2.45	14.6
1	14.06	2.15	2.61	17.6
1	14.83	1.64	2.17	14
1	13.86	1.35	2.27	16
1	14.1	2.16	2.3	18
1	14.12	1.48	2.32	16.8
1	13.75	1.73	2.41	16

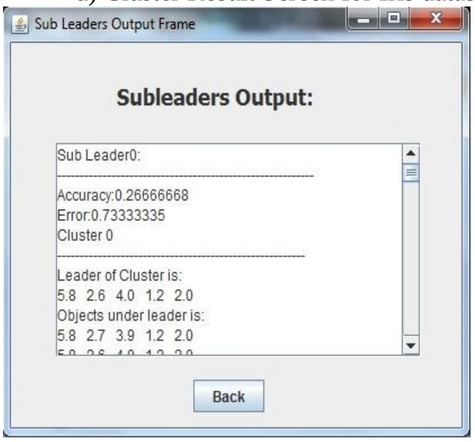
Enter Threshold Value: 54

c) Cluster Result Screen for wine dataset



```
Sub Leader0:
-----------------------------------------------------
Accuracy:0.8333333
Error:0.16666669
Cluster 0
---------------------------------------------------
Leader of Cluster is:
1.0  13.83  1.65  2.6  17.2  94.0  2.45  2.99  0.22  2.2
Objects under leader is:
1.0  13.83  1.65  2.6  17.2  94.0  2.45  2.99  0.22  2.2
```

d) Cluster Result Screen for Iris dataset



```
Sub Leader0:
-----------------------------------------------------
Accuracy:0.26666668
Error:0.73333335
Cluster 0
---------------------------------------------------
Leader of Cluster is:
5.8  2.6  4.0  1.2  2.0
Objects under leader is:
5.8  2.7  3.9  1.2  2.0
5.8  2.6  4.0  1.2  2.0
```

IV. Test Process:

4.1 Object Name: Input File Selector
Test Criteria: Verifies whether the user input is valid or not.
Testing Approach: Black Box Testing

Test case Specification:

Field Name	Input	Expected O/P	Comments
dataset file path	String	Reject	Fail
	Numeric	Reject	Fail
	Alphanumeric	Reject	Fail
	file path	Accept	Pass

4.2 Object Name: Threshold Value Input Frame
Test Criteria: Verifies whether the user input is valid or not.
Testing Approach: Black Box Testing

Test case Specification:

Field Name	Input	Expected o/p	Comments
Number Of Clusters	String	Reject	Fail
	Integer	Accept	Pass
	Float	Reject	Fail
	Alphanumeric	Reject	Fail

4.3 Object Name: Sub Threshold Value Input Frame
Test Criteria: Verifies whether the user input file is in the specified form or not and converts the given document dataset into data vectors.
Testing Approach: Black Box Testing

Test case Specification:

Field Name	Input	Expected O/P	Comments
Outlier Factor Input	String	Reject	Fail
	Integer	Accept	Pass
	Float	Reject	Fail
	Alphanumeric	Reject	Fail

4.4 Object Name: Leaders Algorithm
Test Criteria: Verifies whether the user appropriate dataset and number of clusters or not.
Testing Approach: Black Box Testing

Test case Specification:

Field Name	Input	Expected O/P	Comments
dataset file path	String	Reject	Fail
	Numeric	Reject	Fail
	Alphanumeric	Reject	Fail
	file path	Accept	Pass
Threshold Value	String	Reject	Fail
	Integer	Accept	Pass
	Float	Reject	Fail
	Alphanumeric	Reject	Fail

4.5 Object Name: Sub Leaders Algorithm
Test Criteria: Verifies whether the user data points are gathered successfully or not from the given file into data points
Testing Approach: Black Box Testing

Test case Specification:

Field Name	Input	Expected O/P	Comments
dataset file path	String	Reject	Fail
	Numeric	Reject	Fail
	Alphanumeric	Reject	Fail
	file path	Accept	Pass
Sub Threshold Value	String	Reject	Fail
	Integer	Accept	Pass
	Float	Reject	Fail
	Alphanumeric	Reject	Fail

V. Conclusion

In this work, limitations of the well known clustering techniques for large data sets and the details of the proposed clustering method, Leaders–Subleaders, have been presented. Our experimental results on numerical data sets show that the Leaders–Subleaders algorithm performs well. Hierarchical structure with required number of levels can be generated by the proposed method to find the subgroups/subclusters within each cluster at low computation cost. The representatives of the subclusters help in improving the CA and hence the Leaders Subleaders algorithm performs better than the leader algorithm. In bioinformatics (consisting of sequence data sets), it is required to find the subgroups/subfamilies in each of the protein group/family and

Leaders–Subleaders algorithm can be used for this application. Thus the proposed algorithm can be used in applications requiring clustering of large set of numerical data, sequence data and also on text and web document collections.

References

[1]. Ananthanarayana, V.S., Murty, M.N., Subramanian, D.K., 2001. Efficient clustering of large data sets. Pattern Recognition Lett. 34, 2561–2563.

[2]. Bandera, A., Urdiales, C., Arrebola, F., Sandoval, F., 1999. 2D object recognition based on curvature functions obtained from local histograms of the contour chain code. Pattern Recognition Lett. 20, 49–55.

[3]. Berkhin, P., 2002. Survey of clustering data mining techniques. Accrue software, Technical Report. Available from http://citeseer.nj.nec.com/berkhin02survey.html

[4]. Carpenter, G., Grossberg, S., 1990. ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Networks, 3.

[5]. Duda, R., Hart, P., Stork, D., 2002. Pattern Classification, second ed. John Wiley.

[6]. Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on KDD, Portland, Oregon, pp. 226– 231.

[7]. Guha, S., Rastogi, R., Shim, K., 1998. CURE: An efficient algorithm for clustering large databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Seattle, Washington, pp. 73–84.

[8]. Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Comput. Surveys 31 (3), 264–323.

[9]. Kaufman, L., Rousseeuw, P., 1990. Finding groups in data: An introduction to cluster analysis. John Wiley and Sons, New York.

[10]. Kohonen, T., 1985. Median strings. Pattern Recognition Lett. 3, 309–313.

[11]. Ng, R.T., Han, J., 1994. Efficient and effective clustering methods for spatial data mining. In: Proceedings of 20th International Conference on VLDB, Santiago, Chile, pp. 144–155.