Efficient Clustering Techniques in presence of Noise

V.Venkateswara Rao
Associate Professor,
Dept of Computer Science and Engineering,
Pedatadepalli, Tadepalligudem

Dasu Dasari
Final M.Tech Student
Dept of Computer Science and Engineering,
Pedatadepalli, Tadepalligudem

ABSTRACT
Mining Information and Knowledge patterns from large databases have been recognized by many researchers as key research topic in database systems, Knowledgebase systems and in Information providing services. Clustering analysis method is one of the main analytical methods in data mining; the method of clustering algorithm will influence the clustering results directly. Clustering can be applied on database using various approaches based on distance, density, hierarchy and partition. The presence of Noise is a major problem in clustering. Noise is a data item that is not relevant to data mining. The Objective of the paper is present new algorithms for clustering techniques that handles the noise effectively. Our focus is to show the effect of noise on the performance of various types of clustering techniques and to study how noise affects the clustering process in terms of time and space. We have implemented various clustering techniques such as CURE, KMediods. We have computed time complexity and space complexity of various clustering techniques for different number of clusters. These results are presented in various visual presentations like Line Chart, Bar Chart. Then we will conclude which algorithm is more efficient to deal noise.

Keywords: Knowledge Discovery, Data Mining, Clustering Techniques, Noise, pattern recognition, KMeans, KMediods, PAM, CURE, FCMeans.

1. INTRODUCTION
In recent years data analysis is most important task. The need of generating and collecting data has been increasing rapidly [1][2][3][4][5]. The computerization of business organizations and government organizations has increased the need of data collection and data processing. As a result we have millions of databases used in Business Management, Government Administration, Scientific and

Engineering, Marketing Management, Fraud Detection and in many other applications [1][2][3][4]. Today the number of such databases has increased and keeps growing rapidly. This exclusive growth in Database Systems raises the need of data mining [2][3]. The Data Analysis is becoming more and more important to implement decision support systems. To perform data analysis efficiently and effectively, we need new techniques and tools that intelligently and automatically transforms the processed data into knowledge [1][2][3][4][5][6][7][8][9]. Data Mining also referred to as Knowledge discovery in Databases is a process of extraction of unknown and potentially information from databases[1][2][3][4][5][6]. The Data mining using heuristic searches, Artificial Intelligence and expert systems is more useful to search data from large databases. [1][2][3][4][5].Mining knowledge from large databases has been recognized in the industry as key research topic. The discovered knowledge can be applied to information management, query processing, and decision making process control and in many other applications [2]. The Data mining is also useful in a wide range of profiling practices such as marketing, surveillance, fraud detection and scientific discovery [1].In response to such a demand our paper is to provide an implementation of clustering techniques efficiently with presence of noise in data.

1.1 Knowledge Discovery Process
Data Mining, also popularly known as Knowledge Discovery in Databases (KDD)[1][2][3], refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. The following figure (Figure 1.1) shows data mining as a step in an iterative knowledge discover process.

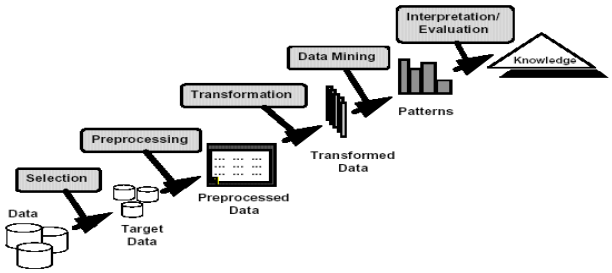

Fig 1.1 (Data mining is the core of Knowledge discovery Process)

The knowledge discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

1) Data cleaning: It also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
2) Data integration: In this stage, multiple data sources, often heterogeneous, may be combined in a common source.
3) Data selection: At this step, the data relevant to the analysis is decided on and retrieved from the data collection.
4) Data transformation: It is also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
5) Data mining: It is the crucial step in which clever techniques are applied to extract patterns potentially useful.
6) Pattern evaluation: In this step, strictly interesting patterns representing knowledge are identified based on given measures.
7) Knowledge representation: This is the final phase in which the discovered knowledge is visually represented to
the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.
This paper is organized as follows: Section 2 completely describes System Design and Implementation. Section 3 describes Proposed work(Clustering techniques to handle noise). Section 4 presents experimental results and

performance analysis. Section 5 presents conclusion and future scope.

2. SYSTEM DESIGN AND IMPLEMENTATION

The overall system design of the System is described in Figure 2.1.

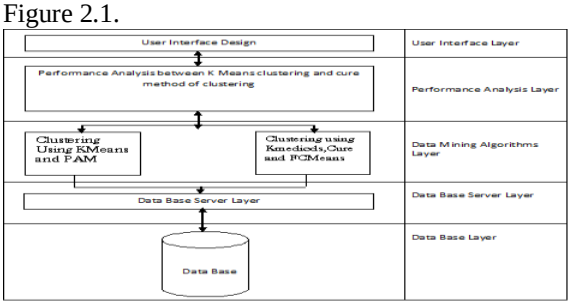

Figure 2.1. System Architecture

The system is composed of five layers

1) User interface layer
2) Performance analysis layer
3) Data mining algorithms implementation layer
4) Data base server layer
5) Data base layer

User Interface Layer:
This layer is responsible for interaction with user and various calls to different graphical and visualization utilities. This module provides an interface for user to invoke with system and to execute queries based clustering algorithms. The following are different services that this module offers.

- To design an interface for CURE clustering
- To design an interface for KMediods clustering
- To generate graph that displays clusters using CURE algorithm
- To generate graph that displays clusters using KMediods algorithm
- Computing time and space required to generate clusters.
- Generating graph showing clusters of KMedoids algorithm
- Generating graph showing clusters of CURE algorithm

Performance analysis layer :
This layer presents the computational complexity and time complexity for generating clusters using Kmeans algorithm and using CURE algorithm. The computational and time complexities are generated for different sizes of data.

Data mining algorithm implementation :
This layer is main layer that connects all system components together. This layer is divided into sub modules .

1) Clustering using K MEDOIDS
2) Clustering using CURE

Database server layer :
This layer uses a database server for maintaining transactional data items for use in K MEDOIDS clustering, and CURE clustering here we are using ORACLE 10G as database server.

Database layer :
This layer is collection of database tables that maintains transactional data items.

3. PROPOSED WORK: CLUSTERING

"The process of organizing objects into groups whose members are similar in some way". Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge of the group definitions. Various Clustering techniques available based on different parameters like distance, density, hierarchy and partition. Popular clustering techniques include k-means clustering based on partitioning using distance, Kmediods also based on partitioning using distance as centroid, Pam based on partitioning using density, Cure based on hierarchical clustering and FCMeans based on Neural networks.
Some basic features of clustering:
• The number of clusters is not known.
• There may not be any prior knowledge concerning the clusters.
• Cluster results are dynamic

3.1 KMediods Clustering Algorithm

K-medoids is a clustering algorithm that is related to the k-means algorithm. The k-medoids is a partitioning algorithm that divides the data set up into separate clusters. The algorithm attempts to minimize the squared error, which is the distance between points in the cluster and a point that is designated as the center (medoid) of a cluster. A medoid is considered an object of a cluster whose average dissimilarity to all the objects in a cluster is minimal. The k-medoids algorithm functions by placing data into k clusters. k is a predetermined number that is chosen before the algorithm is executed. The algorithm functions as follows.
1. Randomly select k objects that will serve as the medoids.
2. Associate each data point with its most similar medoid using a distance measure and calculate the cost.
3. Randomly select a new set k objects that will serve as the medoids and save a copy of the original set.
4. Use the new set of medoids to recalculate the cost.
5. If the new cost is greater than the old cost then stop the algorithm.
6. Repeat steps 2 through 5 until there is no change in the medoids.
The cost for the current clustering configuration is calculated using Equation 1.
Cost (M,X)

$$\sum_{=i=1}^{n} \min\big(d(mj, xi)\big) \qquad (1)$$

Where M is the set of medoids, X is the data set, n is the number of events, k is the number of clusters, m_j is the j^{th} medoid, xi is the ith event, and d is a distance function. The distance function can be any distance metric, Euclidean distance for instance. Equation 1 basically calculates the total cost across the entire data set. The min function is meant to find the medoid that a given event is closest to. This is done by calculating the distance from every medoid to a given event and then adding the smallest distance to the total cost.
.

3.2 CURE Clustering Algorithm

In this paper, we propose a new clustering method named Enhanced CURE (Clustering Using Representatives) to deal noise effect (CURE-NS i.e. CURE with new Shrinking Scheme). CURE employs a novel hierarchical clustering algorithm that adopts a middle ground between the centroid-based and the all-point extremes. In CURE, a constant number c of well scattered points in a cluster are first

chosen. The scattered points capture the shape and extent of the cluster. The chosen scattered points are next shrunk towards the centroid of the cluster by a fraction cr. These scattered points after shrinking are used as representatives of the cluster. The clusters with the closest pair of representative points are the clusters that are merged at each step of CURE's hierarchical clustering algorithm. The scattered points approach employed by CURE alleviates the shortcomings of both the all-points as well as the centroid-based approaches.
It enables CURE to correctly identify the clusters Figure 2(a) - the resulting clusters due to the centroid-based and all-points approaches is as shown in Figures 2(b) and 2(c), respectively. CURE is less sensitive to outliers since shrinking the scattered points toward the mean dampens the adverse effects due to outliers. outliers are typically further away from the mean and are thus shifted a larger distance due to the shrinking. Multiple scattered points also enable CURE to discover non-spherical clusters like the elongated clusters shown in Figure 2(a). For the centroid-based algorithm, the space that constitutes the vicinity of the single centroid for a cluster is spherical. Thus, it favors spherical clusters and as shown in Figure 2(b), splits the elongated clusters. On the other hand, with multiple scattered points as representatives of a cluster, the space that forms the vicinity of the cluster can be non-spherical, and this enables CURE to correctly identify the clusters in Figure 2(a).

CURE's hierarchical clustering algorithm whose salient features are: (1) the clustering algorithm can recognize arbitrarily shaped clusters (e.g., ellipsoidal),
(2) the algorithm is robust to the presence of outliers, and (3) the algorithm has linear storage requirements and time complexity of O(n') for low-dimensional data. The n data points input to the algorithm are either a sample drawn randomly from the original data points, or a subset of it if partitioning is employed.

CURE-NS is an available hierarchical clustering method that is not dependent on the shape of cluster and less sensitive to outliers. Instead of using the centroid as the reference of shrinking, for a point being shrunk, we select a point from its neighborhood as reference. The shrinking procedure can be summarized as follows.

1. A fixed number of the scattered points are randomly chosen from the cluster.
2. Calculate the density distribution of the cluster at the positions of the scattered points that is assigning a density value for each scattered point.
3. The initial reference set is defined as empty set. Choose the point with maximal density value as the first point of the reference set.
4. If a scattered point is added to the reference set, this point is removed from the set of scattered points.
5. Compute the distance between the scattered point I and the reference set R in which d, is the distance between point i and point j that belongs to the reference set.
6. Choose a scattered point i with minimal d_{iR}, and point j is the closest point to i in the reference set. Compare the density values f(i) and f(j) of points i and j to determine which point should be shrunk.
point is shift toward point j , if f (i) < f(j)
both of point i and j do not shift, if f(i) = f (j)
point j shift toward point i, if f(i) > f(j)And the distance of shifting is also dependent on their density values.

$$S(i,j)= \alpha\, d_{ij}\, \frac{|f(i)-f(j)|}{\text{Max}(f(i),f(j)}$$

Where ***α*** is a fraction value.
7. Add the scattered point *i* into the reference set, and re-calculate the position and density value of point ***i*** or *j* due to shrinking.
8.The remaining scattered points are processed as step *5-6* until all points are added into the reference set.c The final reference set is used as the set of representative points.

4. EXPERIMENTAL RESULTS

To establish the practical efficiency of the algorithms, we implemented them and tested its performance on a number of data sets containing students semester marks data. These included both synthetically generated data and data used in real applications. The algorithms were implemented in Java and C# (.Net Framework) and were run on windows xp operating environment. During implementation of these clustering algorithms, we had computed time and space complexities for different no of clusters and for different noise percentages. The Figure 4.1 and 4.2 shows generation of clusters using KMediod's and CURE algorithms. Here the Input to the algorithm is Data Set and No of Clusters and out put is Clusters.

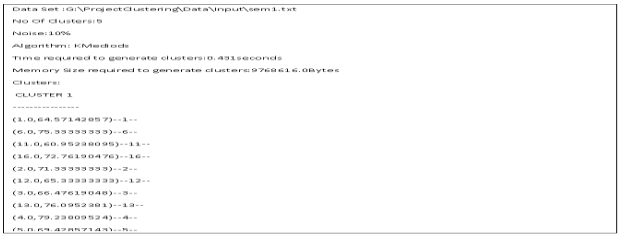

Figure 4.1 Clusters generated by KMediods algorithm

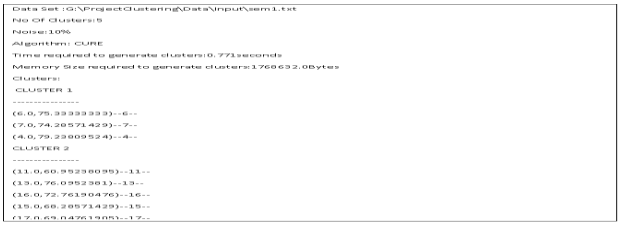

Figure 4.2 Clusters generated by CURE algorithm

The Figure 4.3 shows clusters generated by KMediod's algorithm in Graph

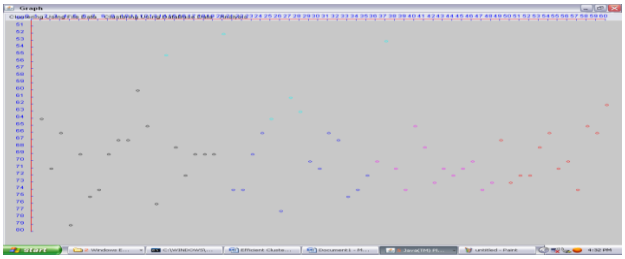

Figure 4.3 Graph of clusters generated by KMediods algorithm.

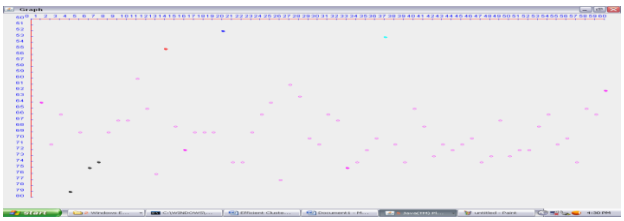

Figure 4.4 Graph of clusters generated by CURE algorithm

Figure 4.5 and 4.6 shows the performance of KMediod's and CURE algorithms.

S. No	Algorithm	No Of Clusters	Noise %	Time Seconds	Memory Size Bytes
1	Kmediods	5	10	0.21	1508568
2	Kmediods	5	20	0.301	1411072
3	Kmediods	5	30	0.301	2077792
4	Kmediods	5	40	0.551	2584976
5	Kmediods	8	10	0.351	1280168
6	Kmediods	8	20	0.361	2132120
7	Kmediods	8	30	0.751	2435688
8	Kmediods	8	40	0.892	2584976

Figure 4.5 Analysis of KMediod's algorithm for different noise values.

The above figure shows as the number of clusters and noise % increases then the time to generate clusters increases.

S.No	*Algorithm*	*No Of Clusters*	*Noise %*	Time Seconds	Memory Size Bytes
1	*CURE*	*5*	*10*	*0.5*	*1401704*
2	*CURE*	*5*	*20*	*0.932*	*1791032*
3	*CURE*	*5*	*30*	*2.314*	*2043664*
4	*CURE*	*5*	*40*	*3.355*	*2581976*
5	*CURE*	*8*	*10*	*2.314*	*1258256*
6	*CURE*	*8*	*20*	*2.934*	*1783008*
7	*CURE*	*8*	*30*	*5.038*	*2348048*
8	*CURE*	*8*	*40*	*7.641*	*3201352*

Figure 4.6 Analysis of CURE algorithm for different noise values.

The above figure shows as the number of clusters and noise % increases then the time to generate clusters increases.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the problems of noise in generating clusters accurately and also we analyzed performance of different clustering algorithms with presence of noise in data.

We have generated clusters and computed the results time complexity and space complexity in the presence of noise for different algorithms. We have concluded KMedoids algorithm takes less time and space compare to CURE.

6. REFERENCES

[1] Amaninder Kaur, Pankaj Kumar , Paritosh Kumar "Effect of Noise on the performance of Clustering Techniques" IEEE International Conference on Networking and Information Technology,2010.

[2] S. Chen, J Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.

[3] JHan and M. Kamber. "Data Mining: Concepts and Techniques". Morgan Kaufinann, 2000.

[4] Piatetsky-Shapiro and W. J. Frawley "Knowledge in Databases". AAAIIMIT Press, 1991.Discovery

[5] A. Jain and R. Dubes. Algorithms for Clustering Data. Prentice Hall, 1998.

[6] P.-N. Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining. Addison-Wesley, 2005.

[7] Domenico Daniele Bloisi and Luca Locchi, Rek Mean ,A k- Means Based Clustering Algorithm

[8] Introduction To Data Mining and Knowledge Discovery,Third Addition ,Two CrowsCorporation. ISBN:I-892095-02-5s.

[9] W. J Frawley, G. Piatetsky-Shapiro and C. J. Matheus, Knowledge Discovery in Databases: An Overview.

[10] T.Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996 .