AN EFFICIENT NEW STRATEGY TO DEFINE CLUSTERING ALGORITHM

Gaurav Gupta*

Himanshu Aggarwal*

ABSTRACT

Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data. This paper presents the concept of data mining and aims at providing an understanding of the overall process and present a clustering algorithm based on new validity index for calculating the value of k in advance. The application of this new k-means algorithm to partition of Sample Training Needs Analysis shows its feasibility and validity. The K - means algorithm is one of the best known and most popular clustering algorithms. K - means seeks an optimal partition of the data by minimizing the sum - of - squared - error criterion. In this paper we present a clustering algorithm based on various validity indices for calculating the value of k in advance. We propose a method for finding cluster center just to overcome the calculation of finding k in k-means algorithm by using various validity measures

Keywords: *Data mining, Clustering, K-means algorithm, Validity index.*

* University College of Engineering, Punjabi University, India.

I. INTRODUCTION

Data mining is the process of discovering useful information (i.e. patterns) underlying the data. Powerful techniques are needed to extract patterns from large data because traditional statistical tools are not efficient enough anymore [1]

The Knowledge Discovery Process steps:

1. Identify business problem
2. Data mining
3. Action
4. Evaluation and measurement
5. Deployment and integration into businesses processes

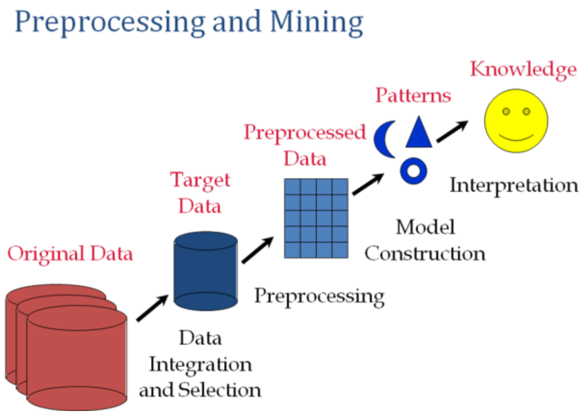


Fig 1.1

Clustering is an important data mining technique that puts together similar objects into a collection in which the objects exhibit certain degree of similarities. Clustering also separates dissimilar objects into different groups. Clustering describes the underlying structure of the data by its unsupervised learning ability. Due to its unsupervised learning ability, it is able to discover hidden patterns of datasets [2]. This has made clustering an important research topic of diverse fields such as pattern recognition, bioinformatics and data mining. Partitioning a set of objects in databases into homogeneous groups or clusters is a fundamental operation in data mining. It is useful in a number of tasks, such as classification (unsupervised), aggregation and segmentation or dissection [4]. The problem of clustering in general deals with partitioning a data set consisting of n points embedded in m-dimensional space into k distinct set of clusters, such that the data points within the same cluster are more similar to each other than to data points in other clusters. The three sub-problems[1] addressed by the clustering process are (i) defining a similarity measure to judge the similarity (or distance) between different elements (ii) implementing an efficient algorithm to discover the clusters of most similar elements in an unsupervised way and (iii) derive a description that can characterize the elements of a cluster in a succinct manner. Traditional clustering algorithms used Euclidean distance measure to judge the similarity of two data elements [5] [6]. This works well when the defining attributes of a data set are purely numeric in nature. However, Euclidean distance measure fails to capture the similarity of data elements when attributes are categorical or mixed. Increasingly, the data mining community is inundated with a large collection of categorical data [3] like those collected from banks, or health sector, web-log data and biological sequence data. Banking sector or health sector data are primarily mixed data containing numeric attributes like age, salary, etc. and categorical attributes like sex, smoking or non-smoking, etc. Clustering mixed data sets into meaningful groups is a challenging task in which a good distance measure, which can adequately capture data similarities, has to be used in conjunction with an efficient clustering algorithm [2].

II. CLUSTERING TECHNIQUES

Different approaches to clustering data can be disc - ribbed with the help of the hierarchy [4] (other axonometric representations of clustering Methodology is possible); ours is based on the discussion in Jain and Daubes. At the top level, there is a distinction between hierarchical and partition approaches (hierarchical methods produce a nested series of partitions, while partition methods produce only one) must be supplemented by a discussion of cross-cutting issues that may (in principle) affect all of the different approaches regardless of their placement in the taxonomy.

Agglomerative *vs.* divisive [3]

This aspect relates to algorithmic structure and operation. An agglomerative approach begins with each pattern in a

Distinct (singleton) cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all patterns in a single cluster and performs splitting until a stopping criterion is met [4].

Monothetic vs. polythetic

This aspect relates to the sequential or simultaneous use of features in the clustering process. Most algorithms are polythetic;

That is, all features enter into the computation of distances between patterns, and decisions are based on those distances. A simple monothetic algorithm reported in Ander berg considers features sequentially to divide the given collection of patterns. This is illustrated in Figure 8.Here, the collection is divided into two groups using feature x1; the vertical broken line V is the separating line. Each of these clusters is further divided independently using feature x2, as depicted by the broken lines H1 and H2. The major problem with this algorithm is that it generates 2d clusters where d is the dimensionality of the patterns. For large values of d (d. 100 is typical in information retrieval applications [Salton 1991]), the number of clusters generated by this algorithm is so large that the data set is divided into uninterestingly small and fragmented clusters.

Hard *vs.* fuzzy [4]

A hard clustering algorithm allocates each pattern to a single cluster during its operation and in its output. A fuzzy clustering Method assigns degrees of membership in several clusters to each input pattern. A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership.

Deterministic *vs.* stochastic [4]

This issue is most relevant to partitioned approaches designed to optimize a squared error function. This optimization

Can be accomplished using traditional techniques or through a random search of the state space consisting of all possible labeling.

Incremental *vs.* non-incremental [4]

This issue arises when the pattern set to be clustered is large, and constraints on execution time or memory space affect the architecture of the algorithm. The early history of clustering methodology does not contain many examples of clustering algorithms designed to work with large data sets, but the advent of data mining has fostered the development of clustering algorithms that minimize the number of scans through the pattern set, reduce the number of patterns examined during execution, or reduce the size of data structures used in the algorithm's operations [4].

III. K-MEANS CLUSTERING: THE ALGORITHM

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem [7]. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid [7]. When no point is pending, the first step is completed and an early group age is done. At this point need to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function [8]. The objective function

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\left\| x_i^{(j)} - c_j \right\|^2$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centers. The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated [8].

The algorithm is significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains [6].

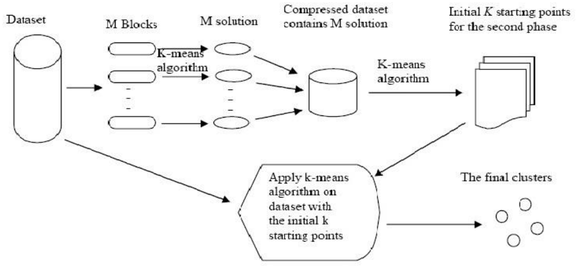
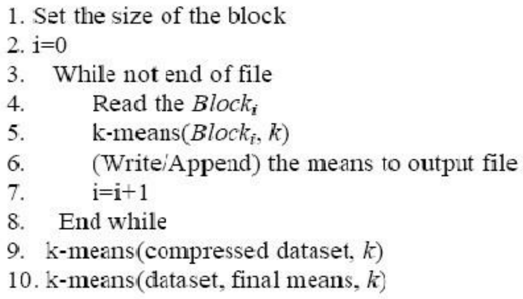

Dataset
M Blocks
M solution
Compressed dataset contains M solution
Initial *K* starting points for the second phase
K-means algorithm
K-means algorithm
Apply k-means algorithm on dataset with the initial k starting points
The final clusters

Fig 3.1



```
Set the size of the block
i=0
While not end of file
    Read the $Block_i$
    k-means($Block_i$, $k$)
    (Write/Append) the means to output file
    i=i+1
End while
k-means(compressed dataset, $k$)
k-means(dataset, final means, $k$)
```

Fig 3.2

IV. WEAKNESS OF K-MEANS CLUSTRING

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly [10].
2. The number of cluster, K, must be determined beforehand.
3. The real cluster never knows, using the same data, if it is inputted in a different order may produce different cluster if the number of data is a few.
4. Weakness of arithmetic mean is not robust to outliers. Very far data from the centroid may pull the centroid away from the real one [10].

V. SOLUTION

Cluster Index[5]: To obtain the natural clusters in the data set the user need to specify the number of desired clusters depending on the actual classes present in data set objects. But it is difficult to determine the number of clusters k, if the user doesn’t have any prior knowledge about the data set objects. A technique to solve this problem is to define a cluster index [9], which is used to estimate the optimal number of clusters. The index used to determine the cluster index which is the ratio of the average intra-cluster compactness to inter-cluster separation.

The best clustering result is the one in which the average intra-cluster variance is minimum and at the same time average inter cluster variance is maximum. Thus the value of k which gives minimum Vxb will decide upon the optimal k*.

1) Input

[kmin, kmax]: a value range of clustering number.

D: a data set containing n objects.

2) Output

An Optimal clustering number k*

3) Algorithm

```
1. set minVxb=_
2. for i=kmin : kmax
    a. run the conventional k-medoids algorithm.
    b. calculate newVxb
    c. if newVxb < minVxb
        i. set minVxb = newVxb
        ii. K*=i
3)Output K*
```

VI. CONCLUSION

In this paper, we have presented k-means algorithm in detail. We have seen that there are various limitations in this algorithm, and to remove the problem of specifying the number of clusters initially we have read one method, that is by taking index to find another attribute for finding the value of k is our future work.

VII. REFERENCES

1. Data Mining: Concepts and Techniques, 2nd Edition, Jiawei Han and Micheline Kamber, Morgan Kauffman.
2. K-means tutorial slides (Andrew Moore)
3. J. Han, M. Kamber, “Data Mining: Concepts and Techniques,” Second Edition, Elsevier Inc., Rajkamal Electric Press, 2006.
4. Ó.R. Zaïane, “Introduction to Data Mining,” CMPUT690 Principles of Knowledge Discovery in Databases, University of Alberta, 2009.
5. J. Gu, X. Chen, J. Zhou, “An Enhancement of K-means Clustering Algorithm” 2009 International Conference on Business Intelligence and Financial Engineering, 2009.
6. J. Xie, Y. Zhang, W. Jiang, ”A K-means Clustering Algorithm with Meliorated Initial Centers and Its Application to Partition of Diet Structures,” International Symposium on Intelligent Information Technology Application Workshops, 2008.
7. Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. (2009). "The Planar k-Means Problem is NP-Hard".
8. Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NP-hardness of Euclidean sum-of-squares clustering".
9. Hamerly, Greg, and Elkan, Charles. "Learning the k in k-means." Retrieved from the World.

10. Zhao, Tong, Nehorai, Arye, and Porat, Boaz. "K-Means Clustering-Based Data Detection and Symbol-Timing Recovery for Burst-Mode Optical Receiver."