International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 247-250

AUTOMATIC INITIALIZATION OF MEANS (AIM): A PROPOSED EXTENSION TO THE K-MEANS ALGORITHM

Samarjeet Borah[1] & M. K. Ghose[2]

K-means is a popular and well known partition based clustering method. It generally shows impressive results even in considerably large data sets. Its computational complexity does not suffer from the size of the data set. The main disadvantage faced in performing this clustering is that the selection of initial means. If the user does not have adequate knowledge about the data set, it may lead to erroneous results. In this paper one statistical based method has been proposed to select the initial means of the clustering process automatically.

Keywords: Cluster, Distance Measure, K-means, Centroid

1. Introduction

There is huge amount of data in the world and it is increasing day by day. Everyday new data are collected and stored in the databases. To obtain implicit meaningful information from the data the requirement of efficient analysis methods [1] arises. If a data set has thousands of entries and hundreds of attributes, it is impossible for a human being to extract meaningful information from it by means of visual inspection only. Computer-based data mining techniques are essential in order to reveal a more complicated inner structure of the data. Such techniques are the clustering solutions which help in extracting information from the large dataset.

2. Clustering

Clustering [2][3][4] is a type of unsupervised learning method in which a set of elements is separated into homogeneous groups. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. The variety of techniques for representing data, measuring similarity between data elements, and grouping data elements has produced a rich and often confusing assortment of clustering methods. Clustering is useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations, including data mining, document retrieval, image segmentation, and pattern classification [5][3].

2.1. Partition Based Clustering Methods

Given a database of n objects, a partition based [5] clustering algorithm constructs k partitions of the data, so that an objective function is optimized. Partition based clustering algorithms try to locally improve a certain criterion. The majority of them could be considered as greedy algorithms, i.e., algorithms that at each step choose the best solution and may not lead to optimal results in the end. The best solution at each step is the placement of a certain object in the cluster for which the representative point is nearest to the object. This family of clustering algorithms includes the first ones that appeared in the Data Mining Community.

The most commonly used are K-means [JD88, KR90][6], PAM (Partitioning Around Medoids) [KR90], CLARA (Clustering LARge Applications) [KR90] and CLARANS (Clustering LARge ApplicatioNS) [NH94]. All ofthem are applicable to data sets with numerical attributes.

2.2. K-means Algorithm

K-means [7] is a well known prototype-based, partitioning clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids. The general algorithm was introduced by Cox (1957), and (Ball and Hall, 1967; MacQueen, 1967) [6] first named it k-means. Since then it has become widely popular and is classified as a partitional or non-hierarchical clustering method (Jain and Dubes, 1988).

The K-means algorithm works as follows:

- Select initial centres of the K clusters. Repeat steps 2 through 3 until the cluster membership stabilizes.
- Generate a new partition by assigning each data to its closest cluster centres.
- Compute new cluster centres as the centroids of the clusters.

[1,2]Department of Computer Science & Engineering, Sikkim Manipal Institute of Technology Majitar, Rangpo, East Sikkim-737136, India.

E-mail: [1]samarjeetborah@gmail.com, [2]mkghose2000@yahoo.com

Let us briefly describe the k-means algorithm. Suppose that a dataset of n data points x_1, x_2, …, x_n such that each data point is in R^d, the problem of finding the minimum variance clustering of the dataset into k clusters is that of finding k points $\{m_j\}(j = 1, 2, \ldots, k)$ in R^d such that

$$\frac{1}{n}\sum_{i=1}^{n}\left[\min_{j} d^2\left(x_i, m_j\right)\right] \quad (1)$$

is minimized, where $d(x_i, m_j)$ denotes the Euclidean distance between x_i and m_j. The points $\{ m_j \}$ (j =1, 2, …, k) are known as cluster centroids. The problem in Eq.(1) is to find k cluster centroids, such that the average squared Euclidean distance (mean squared error) between a data point and its nearest cluster centroid is minimized.

The k-means algorithm provides an easy method to implement approximate solution to Eq.(1). The reasons for the popularity of k-means are ease and simplicity of implementation, scalability, speed of convergence and adaptability to sparse data. The k-means algorithm can be thought of as a gradient descent procedure, which begins at starting cluster centroids, and iteratively updates these centroids to decrease the objective function in Eq.(1). The k-means always converge to a local minimum. The particular local minimum found depends on the starting cluster centroids. The problem of finding the global minimum is NP-complete. The k-means algorithm updates cluster centroids till local minimum is found. Before the k-means algorithm converges, distance and centroid calculations are done while loops are executed a number of times, say I, where the positive integer I is known as the number of k-means iterations. The precise value of I varies depending on the initial starting cluster centroids even on the same dataset. So the computational time complexity of the algorithm is O(nkI), where n is the total number of objects in the dataset, k is the required number of clusters we identified and I is the number of iterations, $k \leq n$, $I \leq n$.

K-mean clustering has many weaknesses:

- When the numbers of data are not so many, initial grouping will determine the cluster significantly.
- The number of cluster, K, must be determined before hand.
- It is quite difficult to know the real cluster, using the same data, if it is inputted in a different order may produce different cluster if the number of data is a few.
- Sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the local optimum.
- It is not possible to know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.
- Weakness of arithmetic mean is not robust to outliers. Very far data from the centroid may pull the centroid away from the real one.
- The result is circular cluster shape because based on distance.

Several extensions to the k-means [11][8] has been developed to overcome the weaknesses and improve the efficiency of the algorithm. According to one method to overcome outliers problem, median can be introduced instead of mean.

3. THE PROPOSED METHOD

3.1. Automatic Initialization of Means

The major problem faced during k-means clustering is the efficient selection of means. It is quite difficult to predict the number of clusters k in prior. The k varies from user to user. As a result, the clusters formed may not be upto mark. The finding out of exactly how many clusters will have to be formed is a quite difficult task. To perform it efficiently the user must have detailed knowledge of the domain. Again the detail knowledge of the source data is also required.

Here we want to introduce our proposed idea trying to make the k-means algorithm a bit more efficient. In this attempt we are trying to make the selection process of the initial means automatic. Here it will not be specified from the user. A simple statistical process will select the set of initial means automatically based on the dataset. We are also applying the new technique in clustering using K-means algorithm. This technique allows determining the number of possible cluster in prior to initialize the K-means method.

3.1.1 Methodology

We represent the data set as $X = \{x_i, i=1, 2... N\}$ which consists of N data objects x_1, x_2, …, x_N, where each object has M different attribute values corresponding to the M different attributes. The value of i-th object can be given by:

$$X_i = \{x_{i1}, x_{i2}, \ldots, x_{iM}\}$$

We assume that the relation xi = xk does not mean that xi and xk are the same objects in the real world database. It means that the two objects has equal values for the attribute set A={al, a1, a2, …, am}. The main objective of our algorithm is to find out the value k automatically in prior to partition the dataset into k disjoint subsets. As the clustering criterion, here we are using the most widely used distance measure sum of square Euclidian distance. The algorithm aims at minimizing the average square error criterion as:

$$E = \frac{1}{n}\sum_{i=1}^{k}\sum_{xeC_i}(x - m_i)^2$$

which is a good measure of the within cluster variation across all the partitions. Thus the average square error criterion tries to make the k-clusters as compact and separated as possible.

Let us assume a set of means M = {m_j, j = 1, 2, …, K} which consists of initial set of means that has been generated by the algorithm based on the dataset. Based on these initial means the dataset will be grouped into K clusters. Let us assume the set of clusters as C = (c_j, j = 1,2,…, M}. In the next phase the means has to be updated.

In our algorithm the distance threshold has been taken as:

$$dx = \mu \pm 1\sigma \qquad (2)$$

where $\mu = \dfrac{\sum_{i=1}^{n} x_i \overline{M_i}}{\sum_{i=1}^{n} X_{ij}}$

and $\sigma = \sqrt{\dfrac{\sum_{i=1}^{n}\left(X_{ij} - \overline{M_{ij}}\right)^2}{n-1}}$

3.2. Why μ ± 1σ?

In probability theory and statistics, the Gaussian distribution is a continuous probability distribution that describes data that clusters around a mean or average. Assuming Gaussian distribution it is known that μ±1σ contain 65% of the population and thus significant values concentrate around the cluster mean μ. Points beyond this may have tendency of belonging to other clusters. We could have taken μ±2σ instead of μ±1σ, but problem with μ±2σ is that it will cover about 90% of the population and as a result it may lead to improper clustering. Some points that are not so relevant to the cluster may also be included in the cluster.

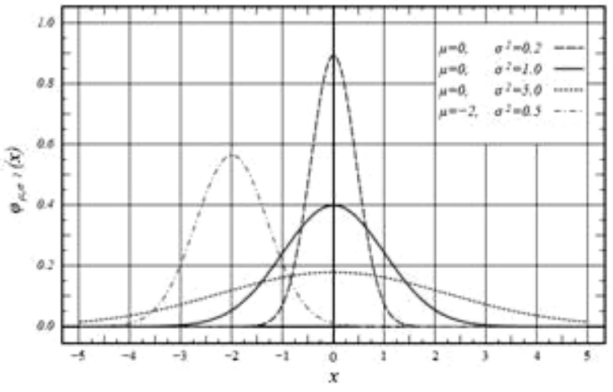

Fig. 1: Normal Distribution

Before the searching of initial means the original dataset D will be copied to a temporary dataset Td. This dataset will be used only in initial set of means generation process. The algorithm INI_MEAN( ) will be repeated for n times (where n is the number of objects in the dataset). The algorithm will select the first mean of the initial mean set randomly from the dataset. Then the object selected as mean will be removed from the temporary dataset. The procedure Cal_dx( ) will compute the distance threshold as given in eq. 1.

Whenever a new object is considered as the candidate for a cluster mean, its average distance with existing means will be calculated as given in the equation below.

$$Adex = \frac{1}{m}\left(\sum_{i=1}^{m} d\left(Marr_i, x_c\right)\right) \qquad (3)$$

where Marr is the set of initial means, i = 1, 2, …, m and m <= n X_c is the candidate for new cluster mean.

If it satisfies the distance threshold then it will considered as new mean and will be removed from the temporary dataset. The process works as follows:

1. Select the initial means:
   a. Find out the mean of each object: $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
   b. Find out the standard deviation (∞) of all objects (only rows will be considered in this case).
   c. Arrange the objects in the ascending order of means.
   d. Select the first object as the first cluster mean.
   e. Select the next mean at the μ–1∞ distance of the first mean.
   f. Again select the next mean at the μ–1∞ distance of the second mean, and so on.
2. Assign rest of the object to the means based on Euclidian distance from the means.
3. Calculate the new means for the 1st level clusters
4. Reassign the points.
5. Repeat step 2 to 4 until criteria function meets

$$E = \sum_{i=1}^{k}\sum_{p \in c_i} |p - m_i|^2$$

The algorithms for the selection of initial set of means are as follows:

Algorithm INI_MEAN (D, Marr)

/* This algorithm will be used to select the initial set of means. It will take the dataset D and the array for initial set of means Marr as inputs. */

1. START
2. Copy D to Td

3. Arbitrarily select x_i as M_1
4. Insert M_1 to Marr
5. Remove x_i from Td
6. dx = Cal_dx(Td)
7. Repeat for i=1 to n
   7.1. Adx= Avg_dist(Marr, x_i)
   7.2. If dx <= Adx, then:
      7.2.1 Remove x_i from Td
      7.2.2 Insert x_i to Marr
      7.2.3 GOTO Step 6
   [End of if structure]

[End of step 6 loop]

8. STOP

Procedure Avg_dist(Marr, x_c, m)

/* This procedure will be used to find out the average distance between the candidate for mean and the rest of the initial set of means already selected. It will take the candidate for mean x_c, array for initial set of means Marr and elements in the Marr m as inputs. */

1. START
2. Set L = 0
3. Repeat for i = 1 to m
   3.1 L=l + d ($Marr_i$, x_c)
   [End of step 3 loop)
4. Return(L/m)
5. STOP

4. CONCLUSION

The most attractive property of the k-means algorithm in data mining is its efficiency in clustering large data sets. It can be shown that the computational complexity of K-means does not suffer from exponential growth with dimensionality rather it is linearly proportional with the number of observations and number of clusters. In this paper we presented a simple idea to enhance the efficiency of k-means clustering by automating the selection of the initial means. From the experiments that that we have made on our proposed scheme we have found that it can improve the cluster generation process of the k-means algorithm, without diminishing the clustering quality in most cases. Our basic idea was to keep the simplicity and scalability of K-means, while achieving automaticity.

REFERENCES

[1] Efficient Classification Method for Large Dataset, Sheng-Yi Jiang School of Informatics, Guangdong Univ. of Foreign Studies, Guangzhou

[2] Clustering Techniques for Larges Data Sets From the Past To the Future, Alexander Hinneburg, Daniel A. Keim

[3] Data Clustering: A Review, A.K. Jain (Michigan State University), M.N. Murty (Indian Institute of Science) and P.J. Flynn (The Ohio State University)

[4] Data Clustering, by Lourdes Perez

[5] Data Clustering and Its Applications, Raza Ali, Usman Ghani, Aasim Saeed.

[6] MacQueen, J. Some methods for classification and analysis of multivariate observations. Proc: 5th Berkeley Symp. Math. Statist, Prob, 1:218-297, 1967.

[7] K-means Clustering: A novel probabilistic modeling with applications Dipak K. Dey(joint work with Samiran Ghosh, IUPUI, Indianapolis), Department of Statistics, University of Connecticut

[8] Variations of K-means algorithm: A tudy for High Dimentional Large Data Sets: Sanjay Garg, Ramesh Ch. Jain, Dept. of Computer Engineering, A. D. Patel Institute of Technlogy, India.

[9] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

[10] Michel R. Anderberg.- Cluster Analysis for Applications: Academic Press, 1973.

[11] Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values ZHEXUE HUANG ACSys CRC, CSIRO Mathematical and Information Sciences, GPO Box 664, Canberra, ACT 2601, Australia.