http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (6): 1132-1135, 2006
ISSN 1812-5638

Variations of k-mean Algorithm: A Study for High-Dimensional Large Data Sets

[1]Sanjay Garg and [2]Ramesh Chandra Jain
[1]Department of Computer Engineering,
A.D. Patel Institute of Technology, New V.V. Nagar, Gujarat State, India
[2]Department of Computer Applications,
Samrat Ashok Technological Institute (Engg. College), Vidisha, Madhya Pardesh, India

Abstract: Clustering algorithms are computationally intensive, particularly when they are used to analyze large amount of high dimensional data. Exploring faster algorithms for clustering is a vital and often encountered research problem. k-mean algorithm is a well known partition based clustering technique and so many variations of this basic algorithm are proposed by various researchers. In order to explore the strength and weaknesses an attempt has been made to compare some of the existing variations of k-mean algorithms using synthetic sets of high dimensional data as benchmark for evaluation and some criteria is also evolved for comparison of clustering algorithms.

Key words: Clustering, data mining, k-mean, high dimensional databases

INTRODUCTION

The clustering problem is a classical problem in the database, knowledge discovery, artificial intelligence and theoretical literature is used to find similar groups of records in very large database. The clustering algorithms are used for similarity search, customer segmentation, pattern recognition, trend analysis and classification. Basically clustering problem is to find a partition of the points into clusters so that the points within each cluster are similar to one another for a given set of points in multidimensional space. Similarity can be determined using various distance functions (Han and Kamber, 2001). Clustering techniques are basically classified in so many categories such as partitioning Techniques, Hierarchical Techmiques, Density based Techniques etc. (Pujari, 1999).

In the past three decades, cluster analysis has been extensively used to many areas such as medicine for classification of diseases, chemistry for grouping of components, social studies for classification of statistical findings, and so on. Its main aim is to discover structures or clusters present in the data. While there is no universal definition of a cluster, algorithms have been developed to find several kinds of clusters: spherical, linear, drawn out, etc. redefining clustering problem in a diverse way for high dimensional applications (Aggarwal and Yu, 2002) facilitates fast computations. Among all the open variations of k-mean clustering algorithm, k-means, k-medoid and h-k-mean clustering algorithm are chosen for our study. Unlike many other partitioning methods, k-means and k-medoid (Kaufmann and Rousseeuw, 1990) are the most basic methods for clustering and h-k-mean (Garg *et al*., 2004) is heuristic based hybrid model of these two algorithm.

The study introduces clustering algorithms based on partition and variations of k-means algorithm i.e., k-mean, k-medoid (PAM) and h-k-mean, presents experimental results comparing the performance of k-means, k-medoid and h-k-mean clustering algorithms on a criterion evolved and finally explains conclusion and future scope in this field.

CLUSTERING ALGORITHMS BASED ON PARTITIONING

Basic concept of Partition based clustering method is to Construct a partition of a database D of *n* objects into a set of k clusters and minimizing an objective function. Exhaustively enumerate all possible partitions into k sets in order to find the global minimum is too expensive. following heuristic is used

- Choose k representations for clusters, e.g., randomly
- Improve these initial representations iteratively
- Assign each object to the cluster it "fits best" in the current clustering
- Compute new cluster representations based on these assignments
- Repeat until the change in the objective function from one iteration to the next drops below a threshold

Corresponding Author: Sanjay Garg, Department of Computer Engineering, A.D. Patel institute of Technology, New V.V. Nagar, Gujarat State, India Tel: +91 2692 236856

The most well-known and commonly used partitioning methods are k-means, k-medoid and their variants.

k-means: The algorithm first select k of the objects, each of which mainly represents a cluster mean or center. Each of the remaining object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then recomputes the new mean for each cluster.

Input: Database of objects D.

Choose k objects as the initial cluster means.
repeat
Assign each object to the cluster to which the object is most similar.
Recalculate the cluster means.
until **convergence criterion is met.**

Fig. 1: Algorithm k-mean

This process iterates until the criterion function converges. Typically the squared-error criterion is used, defined as

$$E=\Sigma_{i=1tok}\Sigma_{p\epsilon Ci} \mid p-m_i \mid^2$$

k-mean (Kuafmann and Rousseeuw, 1990) algorithm is described in Fig. 1.

k-medoid: There are two well-known *k*-medoid methods, PAM and CLARA. PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1990) .To find k clusters, PAM's approach is to determine a representative object for each cluster. This representative object, called a medoid is meant to be the most centrally located object for each cluster. Once the medoids have been selected, each non-selected objects is grouped with the medoid to which it is the most similar. More precisely, if O_j is a non-selected object, and O_i is a (selected) medoid, i. O_j belongs to the cluster represented by O_i, if $d(O_j, O_i) = \min o_e d(Oj, Oi)$, where the notation $\min o_e$ denotes the minimum over all medoids O_e, and the notation $d(O_a, O_b)$ denotes the dissimilarity or distance between objects O_a and O_b. All the dissimilarity values are given as inputs to PAM. Finally, the quality of the chosen medoids is measured by the average dissimilarity between an object and the medoid of its cluster. Figure 2 illustrates k-medoid (Kuafmann and Rousseeuw, 1990) algorithm.

To find the *k* medoids, PAM begins with an arbitrary selection of *k* objects. Then in each step, a swap between a selected object O_i and a non-selected object O_h is made, as long as such a swap would result in an improvement of the quality of the clustering. In particular, to calculate the effect of such a swap between O_i and O_h, PAM computes costs C_{ih} for all non-selected objects O_j. depending on which of the following cases O_j is in, C_{jih} is defined by one of the equation below.

First case: O_j currently belongs to the cluster represented by O_i. Furthermore, let O_j be more similar to $O_{j,2}$ than O_h, i.e., $d(O_j,O_h) >= d(O_j,O_{j.2})$. where $O_{j.2}$ is the second most similar medoid to O_j. Thus, is O_i is replaced by O_h as a medoid, O_j would belong to the cluster represented by $O_{j.2}$. hence, the cost of the swap as far as O_j is concerned is:

$$C_{jih} = d(O_j,O_{j.2}) - d(O_j,O_i). \quad (1)$$

This equation always gives a non-negative C_{jih} indicating that there is a non-negative cost incurred in replacing O_i with O_h.

Input: Database of objects D.
select arbitrarily k representative objects,.
mark these objects as "medoid"
mark the remaining as "non-medoid".
repeat
do for all **medoid objects O_i**
do for all **non-medoid objects O_h**
compute C_{ih}
end do.
end do.

select i_{min}, h_{min} such that $C_{imin,hmin} = Min_{i,h}\ C_{ih}$
if **$C_{imin,hmin} < 0$**
then **mark O_i as non-medoid object and O_h as medoid object**
until **convergence criterion is met.**
find **clusters $C_1, C_2, C_3,\ldots,C_k$.**

Fig. 2: Algorithm k-medoid (PAM)

Second case: O_j currently belongs to the cluster represented by O_i. But this time, O_j is less similar to $O_{j.2}$ than O_h, i.e., $d(O_j,O_h) < d(O_j,O_{j.2})$. Then, if O_i is replaced by O_h, O_j would belong to the cluster represented by O_h. Thus, the cost for O_j is given by:

$$C_{jih} = d(O_j,O_h) - d(O_j,O_i) \quad (2)$$

Unlike in Eq. 1, C_{jih} here can be positive or negative, depending on whether O_j is more similar to O_i or to O_h.

Third case: suppose that O_j currently belongs to a cluster other than the one represented by O_i. Let $O_{j.2}$ be the representative object of that cluster. Furthermore let O_j be more similar to $O_{j.2}$ than O_h. Then even if O_i is replaced by O_h, O_j would stay in the cluster represented by $O_{j.2}$. Thus, the cost is:

$$C_{jih} = 0. \quad (3)$$

Fourth case: O_j currently belongs to the cluster represented by $O_{j.2}$. But O_j is less similar to $O_{j.2}$ than O_h. Then replacing O_i with O_h would cause O_j to jump to the cluster of O_h from that of $O_{j.2}$. Thus the cost is:

$$C_{jih} = d(O_j,O_h) - d(O_j,O_{j.2}) \quad (4)$$

and is always negative. Combining the four cases above, the total cost for replacing O_i with O_h is given by:

$$Tcih = \Sigma Cjih \quad (5)$$

h-k-mean: A hybrid approach of k-mean and k-medoid algorithm is h-k-mean algorithm which can deal with presence of noise and outliers efficiently. Heuristic followed by h-k-mean algorithm (Garg *et al.*, 2004) is as under:

- Initially cluster centers are detected by using the strategy of k-mean algorithm in first iteration
- Cluster centre means are recalculated after temporary removal of most distant object from cluster centre in each cluster and a reference point (medoid) is chosen as new cluster mean for each cluster which is nearest to recalculated mean .Same process is followed for subsequent iterations until algorithm converges.

Running time for this algorithm is a little mre than running time of k-mean algorithm. This algorithm selects reference point which is most centrally located after considering removal of a possible outlier unlike a random reference point in k-medoid algorithm. k-mean algorithm is more robust than both the algorithms i.e., k-mean algorithm and *k*-medoid algorithm in the presence of noise and outliers because farthest values are temporarily removed while deciding cluster representing object.

RESULTS

The simulations were performed on synthetically generated datasets varying their dimension from 2 to 30, varying their size from 1000 points to 100000 points and including zero to 20 numbers of outliers.

Table 1: Summary of comparative study

Criteria of comparison	k-mean algorithm	k-medoid algorithm	h-k-mean algorithm
Average running time	4.352	8.564	5.328
Avg. distance of points in each cluster from representing point/mean	23.65	20.98	19.43
Maximum interclass distance	Maximum	Average	Minimum
Sensitivity to outliers/noise	Yes	Partial#	Partial
Cluster shapes	Convex	Convex+	Convex++
Dependence on initial partitions	More	Less	Less
Average density of clusters	Less	More than k-mean algorithm	More than k-medoid algorithm
Effect of high dimensionality of data	Better than k-medoid algorithm	deficient	Better than other two algorithms
Effect of size of dataset	Better than k-medoid algorithm	Larger dataset Badly affects on running time	Better than other two algorithms

more sensitive than h-k-mean algorithm, +better than k-mean algorithm, ++close to spheres, better than k-medoid algorithm

Criterion used to study k-mean, k-medoid and h-k-mean algorithm is as follows For the computation of each parameter each algorithm has been run on all the datasets(10 times each) and average value/inference is worked out. Ecludian distance function (Han and Kamber, 2001) is used. Experimental findings are summarized in Table 1.

Average running time: Time complexity is always an important criterion for evaluation of an algorithm. Of course these three algorithms are having asymptotically same time complexity but actual running time is different.

Average distance of points in each cluster from representing point/mean: Average distance of points in each cluster from representing point/mean indicates compactness of clusters and it is a one of the quality of cluster parameter.

Maximum interclass distance: Maximum interclass distance (Gonzalez, 1985) is also a quality of cluster parameter and to minimize it always an interesting issue in speculative as well as realistic literature.

Sensitivity to outliers/noise: Outliers and noise present in data badly affects quality of clusters so it always desirous that clustering algorithm should not be sensitive to the noise/outliers.

Cluster shapes: Distance based clustering algorithms generates convex shaped clusters. It is always important to have exactly spherical clusters for various applications viz. indexing applications (Garg and Jain, 2005b).

Average density of clusters: Most of the applications e.g., clustering based indexing for high dimensional databases require dense/compact clusters for better competence. density of cluster can be calculated by dividing number of datapoints in a cluster by area of the cluster.

Dependence on initial partitions: In distance based clustering algorithms initial partitions are made randomly, convergence of this kind of algorithms mostly dependent on initial partitions so it is always desirous to make the algorithm robust to initial partitions.

Effect of high dimensionality of data: Increase in the number of dimensions degrades the performance of clustering algorithm. High dimensional clustering (Garg *et al*., 2005) is required for various applications viz. geographic/spatial information systems (Ng and Han, 1994), gene expression data applications etc. so effect of high dimensions is studied on running time of the algorithms.

Effect of size of dataset: Increase in the size of the dataset crushes the performance of clustering algorithm. Clustering of large datasets is required for various knowledge extraction applications so consequences of size of dataset are extracted out on running time of the algorithms.

DISCUSSION

Numerous variations of most basic clustering algorithm k-means are suggested by many researchers, we studied only k-mean, k-medoid and h-k-mean algorithms for high dimensional large datasets i.e., two most basic algorithms and a hybrid model thereof. This study indicates that in most of the criterion h-k-mean algorithm is overperforming other two. An remarkable observation is that h-k-mean algorithm is providing better quality of clusters even in the presence of outliers and noise. It is also been observed that for large high dimensional dataset h-k-mean algorithm is robust.

Criterion used for comparison may not be sufficient, more criteria may be evolved and a precise mathematical model is required to predict quality of clusters. A high performance clustering algorithm for large high dimensional dataset with variety of attributes is a fundamental and open ended research region.

REFERENCES

Aggarwal, C.C. and P.S. Yu, 2002. Redefining clustering for high dimensional applications. In IEEE Transaction on Knowledge and Data Eng., 14: 210-225.

Garg, S., P. Amit and R.C. Jain, 2004. h-k-mean algorithm: Integrating k-mean and k-medoid clustering algorithm. In Proc. Intl. Conf. on AIECT, Dec 004, 2: 70-76.

Garg, S., P. Amit and R.C. Jain, 2005a. A comparative study of clustering algorithms for high-dimensional Data. J. Eng. Technol. SPU., 18: 120-125.

Garg, S. and R.C. Jain, 2005b. An experimental study on quality of cluster shapes for h-k-mean algorithm. In ADIT J. Eng., 2: 120-125.

Gonzalez, T., 1985. Clustering to minimize the maximum intercluster distance, Theor. Comp. Sci., 38: 293-306.

Han, J. and K. Kamber, 2001. Data Mining: Concepts and Techniques. Morgan Kauffman Publishers.

Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons.

Ng, R. and J. Han, 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. In 20th International Conference on Very Large Data Bases (Bocca, J., M. Jarke and C. Zaniolo, Eds.), (Santiago, Chile), Morgan Kaufmann, pp: 144-155.

Pujari, A.K., 1999. Data Mining Techniques. University Press, Hyderabad.