Modified K-Means Algorithm for Effective Clustering of Categorical Data Sets

M.Ramakrishnan
Professor & Head,
Department of Information Technology, Velammal Engineering College, Chennai-66, Tamil Nadu, India

D.Tennyson Jayaraj
Research Scholar, Department of Computer Science, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India

ABSTRACT
Traditional k-means algorithm is well known for its clustering ability and efficiency on large amount of data sets. But this method is well suited for numeric values only and cannot be effectively used for categorical data sets. In this paper, we present modified k-means algorithms that can that can perform clustering very effectively on mixed data sets. The main intuition behind our proposed method is that all prototypes are the potential candidates at the root level. For the children of the root node, we can prune the candidate set by using simple geometrical constraints. The experimental results show that this method is well suited for categorical data sets and overall time of computation is very minimal.

General Terms
K-means algorithm, Data Mining, Clustering

Keywords
Clustering, Large Data Sets, K-Means algorithm, CLARANS, DBSCAN, Data Mining, Pattern Mining, Rule Mining

1. INTRODUCTION
Forming homogeneous groups from a set of objects in databases is said to be clustering. It is a fundamental operation in data mining and it is useful in classification, aggregation and segmentation processes[1]. Clustering partition a set of objects and each set is said to be one cluster. Objects in a cluster are more similar to each other than objects in different clusters[2].

Clustering methods are classified into hierarchical and partitional. First method proceeds by either merging small clusters into larger ones or by splitting larger clusters where as later method directly attempts to decompose the data set into a set of disjoint clusters

Raymond et al [3] proposed CLARANS (Clustering Algorithm based on Randomized Search) clustering method which aims to use randomized search to facilitate the clustering of larger number of objects. This method is very efficient than PAM (Partitioning Around Mediods) and CLARA (Clustering Large Application) methods.

Martin Ester et al [4] proposed DBSCAN (Density Based Spatial Clustering of Applications with Noise) which discovers the clusters and noise in a spatial database. DBSCAN needs only one input parameters and it supports the user in determining an appropriate value for it.

Zhang et al [5] proposed BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm which is an efficient clustering method for very large databases. This method exploits that data space is not uniformly occupied. A dense region of the points is treated collectively as a single cluster. Moreover, BIRCH makes full use of available memory to derive the finest possible sub clusters [5].

Erich Schikuta [6] proposed GRIDCLUS(Grid Clustering) algorithm which is a fast and efficient hierarchical clustering method for very large data sets. This method uses multi dimensional grid data structure and it is able to deliver structural pattern distribution information for very large data sets. This method can also be used as a connecting element between hierarchical and partitioning methods [6].

Dad Judd et al [7] proposed PCLUSTER (Parallel Clustering) algorithm which is a parallel version of a square error clustering method. This method prunes as much computations as possible while preserving the clustering quality. The major enhancements proposed in this method are computing spheres of guaranteed assignment for centroids, computing maximum movement effect for patterns across iterations and maintaining partial sum of centroids.

Garcia et al [8] proposed clustering method based on dynamic scheme. This method is based on appropriate partitioning with dissimilarity measure between entity pairs as well as splitting by dynamic procedure. With this method, the objects could be separated even without the prior knowledge of given data set. But the computation of dynamic scheme for a data set is more expensive than k-means algorithm.

M.Emre Celebi et al [13] presented a comparative study of efficient initialization method for k-means clustering algorithm. Though k-means clustering algorithm is widely used for many practical applications, initial placement of cluster centers is highly sensitive. Many initialization methods have been proposed to solve this problem. A comparison is made with eight commonly used linear time complexity initialization methods on large and heterogeneous collection of data sets for various performance criteria. The initialization methods analyzed are Forgy's method, Jancey's method, MacQueen's Quick Cluster method, Ball and Hall's method, Simple Cluster Seeking method, Spath's method, Maximin method, Al-Daoud's density based method and Bradley and Fayyad's method.

Michael Ankerst et al [14] proposed OPTICS – Ordering points to identify the clustering structure. Well known clustering algorithms require input parameters that are hard to determine. Moreover, many real-data set does not have a global parameter setting which describes the intrinsic clustering structure accurately. For this purpose, a new algorithm was proposed which creates an augmented ordering of database representing its density-based clustering structure. This structure contains information which is equivalent to density-based clustering corresponding to a broad range of

parameter settings. The cluster ordering for medium sized data sets is represented graphically and for very large data sets, appropriate visualization techniques were introduced. Both are well suited for interactive exploration of the intrinsic clustering structure which offers extra insight into distribution and correlation of data.

K-means algorithm is an effective method of clustering for many practical applications. However, different initial partitions in k-means method results in different final clusters. Moreover, fixed number of clusters in k-means algorithm makes it difficult to predict the value of k. Hence we propose a modified k-means clustering algorithm which is very efficient over traditional method. The performance of the proposed method is higher that the direct k-means algorithm for most of the data sets.

The paper is organized as follows : section 1 provides detail background about the clustering terminology and latest developments. Section 2 describes the basic k-means algorithm. The proposed modified k-means algorithm is extensively discussed in section 3 and section 4 provides experimental results and discussion. Section 5 tells conclusion about the proposed method and paper ends by providing references in section 6.

2. K-MEANS CLUSTERING ALGORITHM

The method defines k centroids, one for each cluster. The best way to place centroids is to place them as much as far possible. Next, associate each given data set to the nearest centroid. When no point is pending, early groupage is performed. New centroids are identified and new binding is done between new centroids and same data set points. This is the looping process and centroids change their location step by step until all the chances are over. Finally the method aims at minimizing an objective function and it is given by

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

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 Cj is the distance of 'n' data points from their cluster centres.

Algorithm k-means
(for partitioning each cluster centre which is represented by mean value)

Input

- k : the number of clusters
- D : a data set containing n objects

Output: A set of k clusters

Method

1) Arbitrarily choose k from D as the initial cluster centres
2) Repeat
3) Reassign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster ;
4) Update the cluster means i.e. calculate the mean value of the objects for each cluster;
5) Until no change

The time required to reassign each object to the cluster is O(nkd). The time required for calculating the centroids (step 3) is O(nd) and for calculating the error function (step 4) is also O(nd). The computational time of k-means algorithm is directly proportional to the number of iteration. However, this time can be reduced if we reduce the number of iterations.

3. PROPOSED METHOD

The two strategies that we exploit in the proposed method which reduces the overall computational time are : using previous iteration information to reduce the number of distance calculations and organizing the data sets in a suitable data structure so that closest data set becomes more efficient. Prototypes are potential candidates for the closest prototype at root level. We can prune childrens of the root node by simple calculations. This step is repeated recursively the size of the candidate set becomes one for each node. At this step, all the patterns of the tree have sole candidate as their closest prototype. By doing this, number of iterations needed for distance calculation can be minimized.

Improvements can be achieved by applying pruning methods. This can be done by using the following steps:-

- Find minimum and maximum distances for each candidate in the sub space.
- Find minimum of the maximum distances (Minmax)
- Prune out all candidates with minimum distance greater than Minmax.

The idea is to organize the closest pattern vectors efficiently. To perform this, we build k-d tree that can organize pattern vectors, root representing all the patterns and child nodes representing subsets of patterns completely contained in sub spaces. Each node in the tree contains information such as the number of points (m), linear sum of points (LS) and square sum of the points (SS). The k-d tree is built with the following choices :

- Using common dimensions for all the nodes at the same level of the tree. Another option is to use splitting dimensions. Here dimensions are chosen in round robin fashion.
- The dimensions can be splitted in to two equal parts. Another option is to divide the dimensions so that equal number of patterns exists on both the sides. It is observed that first approach costs higher than the later one.

Splitting along the longest dimension and choosing mid-point based approach for splitting is preferable. The next step is to derive initial prototypes. This can be done by random method.

For each iteration, k-d tree is traversed by using depth-first approach, starting from root node to all prototypes. Pruning function is applied to each node. If the number of candidate prototypes is equal to one, the traversal below that internal node is not pursued. The cluster information is constantly updated about the points, LS and SS. K-means algorithm is applied on the leaf node if there is more than one candidate prototype.

This approach is more conservative and this may miss few of the pruning opportunities. However, this approach is relatively inexpensive and shows that computation time is proportional to k. If a more expensive pruning algorithm is

chosen, it may decrease the overall number of distance calculations. Whenever iteration ends, the new set of centroids is derived and error function is calculated. This can be calculated by using the formula

$$\sum_{i=1}^{k}\left(C_{ss}^{i} - \frac{\left(\overline{C_{LS}^{i}}\right)^{2}}{C_{n}^{i}} \right)$$

The formula for calculating maximum and minimum distances for each prototype is to use the below formula for each node of the tree independently.

$$\max = \sqrt{\sum_{j=1}^{d}\left(P_{ij} - dist_{ij}\right)^{2}}$$

Where Pij is the current prototype, 'dist' is the corner for prototype i(Pi) and it is computed as follows :

$$dist_{ij} = \begin{cases} B_j^l : \left|B_j^l - P_{ij}\right| > \left|B_j^u - P_{ij}\right| \\ B_j^u : otherwise \end{cases}$$

The coordinates of the child node is exactly the same as parent. The difference is the one dimension which is used for splitting at the parent node. The computation cost is O(1) for each prototype by using the above approach and the overall computational requirement is O(k) for a node with k prototypes.

4. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed method, this algorithm is applied on several data sets. The results are compared with traditional k-means algorithm. A direct comparison of a method with other algorithm is not a good one due to unavailability of their data sets and software.

The quality measure that is used for comparisons are FRD (Factor Reduction in Distance), FRT (Factor Reduction in overall execution Time) and ADC (Average number of Distance Calculations per pattern). Several data sets are used to study the computational aspects of the proposed method. The purpose of using different data sets is that this gives the scaling properties of the method by changing n and k values. We use three data sets D1, D2 and D3 which were used by T.Zhang et al [11]. The experiment is conducted with leaf sizes as 16 and 64. It is observed that the leaf size of 64 yields optimal results. The overall performance is not delicate with the leaf size.

The below table represent the experimental result. It shows the performance of our method in FRD, FRT and ADC. These measures are represented for each the iteration. The overall result is compared with traditional k-means algorithm. It is observed that ADC is very small and range from 0.5 to 1.83 based on the data set and clusters required.

Data Set	**K**	**K-Means Algorithm**	**Total Time**	**FRT**	**FRD**	**ADC**
DS1	16	6.996	2.366	4.916	27.546	0.9704
DS2	16	6.936	2.256	5.196	35.326	0.8204
DS3	16	6.860	2.226	5.236	36.536	0.8104
R1	16	9.616	2.746	5.486	18.676	1.2804
R2	16	18.276	3.986	6.416	28.236	0.9504
R3	16	8.746	2.146	6.966	99.516	0.5004
R4	16	16.946	3.606	6.706	53.876	0.6504
R5	16	16.416	3.366	7.046	15.476	1.4904
R6	16	32.056	5.336	7.816	18.626	1.2904
DS1	64	23.876	3.096	11.12	55.576	1.5204
DS2	64	23.736	3.186	10.66	44.106	1.8304
DS3	64	24.036	3.196	10.75	53.756	1.5604

5. CONCLUSION

We have presented a modified k-means algorithm in this paper for effective clustering process. Experimental results show that our algorithm is improving the efficiency of the clustering method to two orders of magnitude in the total number of distance calculations and the overall time of computation. The intuition is that the earlier iterations can provide some partial clustering information. This information can be potentially used to construct tree such that pruning is more effective. Optimizations that are used in our method can further reduce the number of distance calculations.

6. REFERENCES

[1] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data, Montreal, Canada, pages 103–114, June 1996.

[2] Chien, L.J., Chang, C.C. and Lee, Y.J., "Variant methods of reduced set selection for reduced support vector machines", Journal of Information Science and Engineering , Vol. 26 (1), 2010.

[3] Chien Cung, Chang, and Yuh-Jye Lee, " Generating the reduced set by systematic sampling", Lecture Notes in Computer Science, Vol. 3177, 2004.

[4] Emre C Oomak , Ahmet Arslan, "A new training method for support vector machines: Clustering k-NN support vector machines", Expert Systems with Applications, Vol. 35, pp. 564–568, 2008.

[5] Gowda, K.C. and Diday, E:, "Symbolic clustering using a new dissimilarity measure", Pattern recognition Letters. Vol. 24 (6), pp.567-578, 1991.

[6] Hastie, T., Tibshirani, R., and Friedman, J., The Elements of statistical learning, 2nd edition, Springer, 2008

[7] He, Z., Xu, X. and Deng, S., "A cluster ensemble for clustering categorical data", Information Fusion, Vol. 6, pp. 143-15, 2005.

[8] Huang, Z., "Clustering large data sets with mixed numeric and categorical values", Proceedings of The First Pacific Asia Knowledge Discovery and Data Mining Conference , Singapore, 1997.

[9] Huang, Z., "Extensions to the k-means algorithm for clustering large data sets with categorical values", Data Mining and Knowledge Discovery, Vol. 2, pp. 283-304, 1998.

[10] Huang, Z., "A note on k-modes clustering", Journal of Classification , Vol. 20, pp. 257-26, 2003.

[11] Huang, C.M., Lee, Y.J., Lin, D.K.J. and Huang, S.Y. , "Model selection for support vector machines via uniform design", A Special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis , Vol. 52, pp. 335-346, 2007.

[12] Hsu, C.W., C.C. Chang and C.J. Lin, "Practical guide to support vector classification", Department of Computer Science and Information Engineering National Taiwan University,2003).

[13] M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela, "A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm", Journal of Expert Systems with Applications, 40 (2013) 200-210, September 2012

[14] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, J&g Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", SIGMOD '99 Proceedings of the 1999 ACM SIGMOD international conference on Management of data , Volume 28 Issue 2, June 1999 Pages 49-60