Journal of Computer Science 6 (1): 60-66, 2010
ISSN 1549-3636

Single Pass Seed Selection Algorithm for k-Means

[1]K. Karteeka Pavan, [2]Allam Appa Rao, [3]A.V. Dattatreya Rao and [4]G.R. Sridhar
[1]Department of Computer Applications,
Rayapati Venkata Ranga Rao and Jagarlamudi Chadramouli College of Engineering, Guntur, India
[2]Jawaharlal Nehru Technological University, Kakinada, India
[3]Department of Statistics, Acharya Nagarjuna University, Guntur, India
[4]Endocrine and Diabetes Centre, Visakhapatnam, Andhra Pradesh, India

Abstract: Problem statement: The k-means method is one of the most widely used clustering techniques for various applications. However, the k-means often converges to local optimum and the result depends on the initial seeds. Inappropriate choice of initial seeds may yield poor results. k-means++ is a way of initializing k-means by choosing initial seeds with specific probabilities. Due to the random selection of first seed and the minimum probable distance, the k-means++ also results different clusters in different runs in different number of iterations. **Approach:** In this study we proposed a method called Single Pass Seed Selection (SPSS) algorithm as modification to k-means++ to initialize first seed and probable distance for k-means++ based on the point which was close to more number of other points in the data set. **Result:** We evaluated its performance by applying on various datasets and compare with k-means++. The SPSS algorithm was a single pass algorithm yielding unique solution in less number of iterations when compared to k-means++. Experimental results on real data sets (4-60 dimensions, 27-10945 objects and 2-10 clusters) from UCI demonstrated the effectiveness of the SPSS in producing consistent clustering results. **Conclusion:** SPSS performed well on high dimensional data sets. Its efficiency increased with the increase of features in the data set; particularly when number of features greater than 10 we suggested the proposed method.

Key words: Clustering, k-means, k-means++, local optimum, minimum probable distance, SPSS

INTRODUCTION

Clustering is the process of grouping similar data into groups called clusters, so that the objects in the same cluster are more similar to each other and more different from the objects in the other group (Ankerst *et al*., 1999). It is a useful approach in data mining processes for identifying hidden patterns and revealing underlying knowledge from large data collections (Jain and Dubes, 1988). The cluster analysis is the most fundamental technique in various applications (Berkhin, 2002) such as data mining and knowledge discovery (Fayyad *et al*., 1996), data compression and vector quantization (Gersho and Gray, 1992), pattern recognition and pattern classification (Duda and Hart, 1973; Duda *et al*., 2001), statistics and bioinformatics (Eisen *et al*., 1995; Jiang *et al*., 2005). Existing clustering algorithms can be broadly classified into hierarchical (Jiang *et al*., 2005), partitional, density based (Ester *et al*., 1996), model based and so on (Kaufman and Rousseeuw, 1990). The k-means is the most popular partitional clustering technique for its efficiency and simplicity in clustering large data sets (Lloyd, 1982; MacQueen, 1967). The k-means was voted as one of the top ten algorithms in data mining (Wu et al., 2008). Although the k-means method has a number of advantages over other data clustering techniques, it also has drawbacks as follows. It converges often at a local optimum (Anderberg, 1973), the final result depends on the initial starting centers and the number of clusters. Several variants of the k-means algorithm have been proposed to improve efficiency or accuracy. Improved efficiency generally accomplished by either reducing number of iterations required for final solution or reducing the total number of distance calculations. Therefore selecting a good set of initial seeds is very important. Many researchers introduce some methods to select good initial centers (Bradley and Fayyad, 1998; Deelers and Auwatanamongkol, 2007). Recently, Arthur and Vassilvitskii (2007) propose k-means++- a careful seeding for initial cluster centers to improve clustering

Corresponding Author: K. Karteeka Pavan, Department of Computer Applications, RVR and JC College of Engineering, Chowdavaram, Guntur (District) Andhra Pradesh, India

results. k-means++ is a way of initializing k-means by choosing initial seeds with specific probabilities and is O (log k) competitive. The k-means++ selects first centroid and minimum probable distance that separates the centroids at random. Therefore different results and different number of iterations are possible in different runs. To obtain good results in less number of iterations the k-means++ has to be run number of times. In this study we propose a method, Single Pass Seed Selection (SPSS) algorithm to initialize first seed and the minimum distance that separates the centroids for k-means++ based on the point which is close to more number of other points in the data set. We have evaluated its performance by applying on various datasets and compare with k-means++. The experiments indicate that the SPSS algorithm converge k-means in less number of iterations with unique solution and also it performs well on high dimensioned data sets compared to k-means++.

Related work: k-means is a widely used clustering technique because of its simplicity, efficiency and observed speed and the Lloyds method remains the most popular approach in practice (Lloyd, 1982). It has the drawbacks as (1) A priori fixation of number of clusters (2) Random selection of initial seeds. Inappropriate choice of number of clusters (Pham *et al*., 2004) and bad selection of initial seeds may yield poor results and may take more number of iterations to reach final solution. In this study we are concentrating on selection of initial seeds that greatly affect the quality of the clusters, the number of iterations and number of distance calculations required for final solution. Fahim *et al*. (2006) proposed a method to minimize the number of distance calculations required for convergence. Here we briefly present previous initialization schemes. One of the first schemes of centroids initialization was proposed by Ball and Hall (1967). A similar approach is also provided by Tou and Gonzales under the name Simple Cluster Seeking (SCS) (Tou and Gonzales, 1977) and is adopted in the FACTCLUS procedure. The SCS method is as follows:

- Initialize the first cluster centroid with the first input
- Select a point as a new seed if it is d distance apart from the all selected seeds. Stop when k seed clusters are initialized
- After scanning all input samples, if there are less than k seed clusters generated and then decrease d and repeat 1-2

The SCS and the method suggested by Ball and Hall are sensitive to the parameter d and the presentation order of the inputs.

Astrahan (1970) suggested using two distance parameters, d1 and d2. The method first computes the density of each point in the dataset, which is given as the number of neighboring points within the distance d1 and it then sorts the data points according to decreasing value of density. The highest density point is chosen as the first seed. Subsequent seed point are chosen in order of decreasing density subject to the condition that each new seed point be at least at a distance of d2 from all other previously chosen seed points. This step is continued until no more seed points can be chosen. Finally, if more than k seeds are generated from the above approach, hierarchical clustering is used to group the seed points into the final k seeds. The drawback in this approach is that it is very sensitive to the values of d1 and d2 and requires hierarchical clustering. In the worst case it requires (n2 log n) time complexity.

Kaufman and Rousseeuw (1990) introduced a method that estimates the density through pairwise distance comparison and initializes the seed clusters using the input samples from the areas with high local density. A notable drawback of the method lies in its computational complexity. Given n input samples, at least n(n-1) distance calculation are required. This could be much more time consuming than k-Means itself when n is large.

Katsavounidis *et al.* (1994) suggested a parameter less approach, which is called as the KKZ method based on the initials of all the authors. KKZ chooses the first centers near the “edge” of the data, by choosing the vector with the highest norm as the first center. Then, it chooses the next center to be the point that is farthest from the nearest seed in the set chosen so far. This method is very inexpensive (O (kn)) and is easy to implement. It does not depend on the order of points and is deterministic by nature as single run suffices to obtain the seeds. However, KKZ is sensitive to outliers, since it is selecting farthest point from the selected centroids.

Bradley and Fayyad (1998) proposed an initialization method that is suitable for large datasets. The main idea of their algorithm is to select m subsamples from the data set, apply the k-means on each subsample independently, keep the final k centers from each subsample provided that empty clusters are not be allowed, so they obtain a set contains mk points. They apply the k-means on this set m times; at the first time, the first k points are the initial centers. At the second time, the second k points are the initial centers and so on. And the algorithm returns the best k centers from this set. They use 10 subsamples from the data set, each of size 1% of the full dataset size. Finally, a last round of k-means is performed on this dataset and the cluster centers of this round are returned as the initial seeds for the entire dataset. This method generally performs better than k-means and converges to the local optimal faster. However, it still depends on the random choice of the subsamples and hence, can obtain a poor clustering in an unlucky session.

More recently, Arthur and Vassilvitskii (2007) proposed the k-means++ approach, which is similar to the KKZ (Katsavounidis *et al*., 1994) method. However, when choosing the seeds, they do not choose the farthest point from the already chosen seeds, but choose a point with a probability proportional to its distance from the already chosen seeds. In k-means++, the point will be chosen with the probability proportional to the minimum distance of this point from already chosen seeds. Note that due to the random selection of first seed and probabilistic selection of remaining seeds, different runs have to be performed to obtain a good clustering.

MATERIALS AND METHODS

In this study we will first introduce the k-means and k-means++ algorithms. The k-means method is simple and fast, that works as follows:

- Arbitrarily choose k initial seeds
- Assign each object to the group that has the closest centroid
- Recalculate the positions of the centroids
- Repeat steps 2 and 3 until the positions of the centroids no longer changes

k-means begins with an arbitrary set of cluster centers. k-means++ is a specific way of choosing these centers. The k-means++ is as follows:

Choose a set C of k initial centers from a point-set (x_1, x_2,..,x_n):

1. Choose one point uniformly at random from (x_1, x_2,..,x_n) and add it to C
2. For each point x_i, set $D(x_i)$ to be the distance between xi and the nearest point in C
3. Choose a real number y uniformly at random between 0 and $D(x_1)^2+D(x_2)^2+...+D(x_n)^2$
4. Find the unique integer i so that
5. $D(x_1)^2+D(x_2)^2+...+D(x_i)^2> = y > D(x_1)^2+D(x_2)^2+...+ D(x_{(i-1)})^2$
6. Add x_i to C
7. Repeat steps 2-5 until k centers

Although the k-means++ is O(log k) competitive in worse on all datasets, it also produces different clusters in different runs due to steps 1 and 3 in the algorithm. We propose a method for the steps 1, 3 of k-means ++ to produce unique solution instead of different solutions, rather the proposed method-SPSS algorithm is a single pass algorithm:

For step 1: Initialize the first centroid with a point which is close to more number of other points in the data set.

For step 3: Assume that n(total number of points) points are distributed uniformly to k (number of clusters) clusters then each cluster is expected to contains n/k points. Compute the sum of the distances from the selected point (in step1) to first n/k nearest points and assume it as y.

The SPSS algorithm: Choose a set C of k initial centers from a point-set (x_1, x_2,..,x_n). where k is number of clusters and n is number of data points:

1. Calculate distance matrix Dist in which Dist (i,j) represents distance from i to j
2. Find Sumv in which Sumv (i) is the sum of the distances from ith point to all other points.
3. Find the point i which is min (Sumv) and set Index = i
4. Add First to C as the first centroid
5. For each point x_i, set D (x_i) to be the distance between x_i and the nearest point in C
6. Find y as the sum of distances of first n/k nearest points from the Index
7. Find the unique integer i so that
8. $D(x_1)^2+D(x_2)^2+...+D(x_i)^2> = y>D(x_1)^2+D(x_2)^2+...+ D(x_{(i-1)})^2$
9. Add x_i to C
10. Repeat steps 5-8 until k centers

The matlab code for step 1 and for step 3 of k-means++, for the proposed method, SPSS is as follows:

```
function [ind,y,dist]=findminobject(Data,k)
[n,l] = size(Data)
%%calculate distance matrix in which dist(i,j)
%represents distance from i to j
dist = squareform(pdist(Data,'euclidean'))
%%find sum of the distances from a point to all other
%points
sumv = zeros(n,1)
```

```
for I = 1:n
    for j = 1:n
    sumv(i,1) = sumv(i,1)+dist(i,j)
    end
end
%%find the point which is close to more number of %
other points by finding minimum of sumv
[min1, ind] = min (sumv)
%%Compute y
minv = sort(dist(ind,:))
s = 0
for i = 1: ceil(n/k)% find the sum of distances of first
%n/k nearest points
    s = s+minv(i)
end
y = s
```

The k-means++ algorithm, to select i in the step 5 it may have to repeatedly select y in step 3 and in worse it may takes max $(D(x_1)^2+D(x_2)^2+...+D(x_n)^2)$ passes whereas the SPSS assumed y in single pass. Therefore the SPSS is a single pass algorithm with unique solution while the k-means++ is not.

To assess the quality of the clusters, we used the silhouette measure proposed by Dembele and Kastner (2003); Rousseeuw and Silhouttes (1987). Silhouette measure is used to assess the Quality of the clusters, the measure proposed by Rousseeuw and Silhouttes (1987). Silhouette of an object i, is:

$$s(i) = (bi - ai) / \max(ai, bi) \qquad (1)$$

In the above equation:

- a_i = Average distance of an object i, to other objects in the same cluster
- b_i = The average distance of an object i, to the other objects in the nearest neighbor clusters

The silhouette value lies between −1 and +1. When its value is less than zero, the corresponding object is poorly classified.

Data sets: We have conducted tests on well known Iris, Serum, Ionosphere, Breast Cancer, *E. coli*, Lung Cancer, rocks and mines, Parkinsons, Glass and Wine data bases.

Serum: This data set is described and used in (Iyer *et al*., 1999). It can be downloaded from: http://www.sciencemag.org/ feature/ data/984559.shl and corresponds to the selection of 517 genes whose expression varies in response to serum concentration in human fibroblasts and contains 10 clusters.

Iris: This data set is downloaded from (Aha, 1987). This is the best known database found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day (Pham *et al*., 2004). The data set contains 3 classes (setosa, virginica, versicolor) of 50 instances each, total 150, where each class refers to a type of iris plant (Duda *et al*., 2001).

Ionosphere: Data set contains 351 points with 34 attributes and is classified into 2 clusters.

Breast cancer: The data set contains 569 objects in 30 dimensions and has 2 clusters.

***E. coli*:** The set contains 336 points with 7 attributes and classified into 8 clusters

Glass: Data set contains 214 points with 7 attributes and classified into 6 clusters.

Lung cancer: Data set contains 27 instances with 56 dimensions and has 3 clusters.

Rocks and mines: Data set contains 208 objects in 60 dimensions and classified into 2 clusters.

Parkinsons: This data set contains 195 instances, 22 attributes and 2 clusters.

Wine: The data set contains 178 points and 13 attributes with 3 clusters.

All these are down loaded from UCI Machine learning repository (Aha, 1987), except Serum.

All data sets were normalized in such a way that every point has an average expression value of zero and a standard deviation equal to 1.

RESULTS

We have compared the SPSS with k-means++, by comparing number of iterations required by k-means to reach final solution. We have applied on more than 10 different data sets. On each data set, we have run the k-means++ 20 times and tabulated in Table 1. SPSS results are shown in the Table 2 and a comparative analysis presented in Table 3. According to Silhoutte measure the observed quality of produced clusters from k-means++ and SPSS are also tabulated in Table 4 and Table5.

In case of serum the k-means++ requires at minimum 11 and at maximum 21 iterations in 20 runs. Notice that the SPSS takes 10 iterations to reach the final solution. Coming to the data set lung cancer the

SPSS converges within 3 iterations whereas the k-means++ takes 6, 5, 4, 3 different iterations in different runs. In 20 runs the figure never less than the 3, which is the SPSS takes.

Observe that for the Rocks and mines data set the SPSS takes 10 iterations to reach the final solution whereas k-means++ takes at maximum 15 and at minimum 8, which was occurred only once in 20 runs. k-means++ takes more than 10 iterations in different 18 runs to reach final solution.

When SPSS applied on ionosphere dataset, it takes 7 iterations to produce results. k-means++ initialization may provide better centroids, but it happened only once (in 15th run) in 20 runs see the details in the Table 1. In case of Breast cancer data set, notice that, after selecting initial seeds with the SPSS, k-means takes 7 iterations to converge the solution. We can see that k-means++ takes more than or equal number of iterations in 19 runs and only once it takes less than 7.

Table 1: Number of iterations required by k-means after initial seed selection with the k-means++

Run no.	Breast cancer	*E. coli*	Glass	Ionosphere	Iris	Lung cancer	Parkinson	Rocks mines	Serum	Wine
1	7	10	11	7	4	5	4	15	11	10
2	13	11	50	7	6	3	8	15	17	7
3	8	38	16	7	6	3	3	9	11	7
4	7	10	5	8	5	4	4	12	14	10
5	9	10	16	7	7	4	4	11	19	9
6	11	20	8	7	5	4	6	15	11	10
7	10	21	16	7	7	3	4	11	17	13
8	13	21	10	7	5	6	4	13	11	11
9	15	13	7	8	6	4	8	15	14	6
10	8	15	8	7	6	3	6	14	19	11
11	12	12	10	7	6	5	3	15	11	8
12	6	24	14	7	6	4	4	13	17	9
13	8	13	10	8	9	3	4	16	11	6
14	10	16	10	7	7	4	3	8	14	6
15	15	38	9	6	6	4	3	15	19	11
16	14	24	19	8	7	4	4	15	14	11
17	15	20	8	7	5	3	4	11	11	10
18	14	22	7	7	6	5	4	12	21	10
19	9	15	8	8	7	4	4	13	15	10
20	12	13	11	7	4	3	4	12	14	6

Table 2: Number of iterations required by k-means after initial seed selection with the SPSS

Breast cancer	*E. coli*	Glass	Ionosphere	Iris	Lung cancer	Parkinson	Rocks and mines	Serum	Wine
7	15	11	7	5	3	4	10	10	10

Table 3: Comparative analysis between k-means++ and SPSS

Data set, size and no. of clusters in the population as given in the UCI Machine learning repository	Initializing seeds with k-means++ -comparison with the SPSS in 20 runs				Initializing seeds with the SPSS
	Frequency of more and equal no. of iterations	Frequency of less no. of iterations	Maximum iterations	Minimum iterations	No. of iterations taken by k-means
Breast cancer, 569×30, 2	17+2 = 19	1	15	6	7
E. coli, 336×7, 8	10+2 = 12	8	38	10	15
Glass, 214×7, 6	6+2 = 8	12	50	7	11
Ionosphere, 351×34, 2	5+14 = 19	1	8	6	7
Iris, 150×4, 3	14+4 = 18	2	9	4	5
Lung cancer, 27×56, 3	13+7 = 20	-	6	3	3
Parkinsons, 195×22, 2	4+12 = 16	4	8	3	4
Rocks and mines, 208×60, 2	18+0 = 18	2	16	8	10
Serum, 517×13, 10	20+0 = 20	-	21	11	10
Wine, 178×x13, 3	6+5 = 11	9	13	6	10

Table 4: Variation of clusters qualities measured using Silhoutte (in percentages) in 20 runs after initial seed selection with the k-means++

Breast cancer	*E. coli*	Glass	Ionosphere	Iris	Lung cancer	Parkinson	Rocks and mines	Serum	Wine
68-70	38-47	71-76	43-45	77-80	38-42	71-77	65-70	27-33	67-70

Table 5: clusters quality measured using Silhoutte (in percentages) after initial seed selection with the SPSS

Breast cancer	*E. coli*	Glass	Ionosphere	Iris	Lung cancer	Parkinson	Rocks and mines	Serum	Wine
70	45	75.5	45	80.5	38	74.47	70	31	69.17

For the dataset *E. coli*, observe that the SPSS method of initialization takes 15 iterations to reach the final solution, where as k-means++ produce different results in different runs. In 20 runs less than 15 iterations can be seen in 8 cases and more than or equal to 15 iterations can be found in 9 cases.

When k-means++ applied on iris data set the k-means takes at minimum 4 iterations and at maximum 7 iterations in all 20 runs. According to the SPSS the k-means requires 5 iterations to reach final solution. Though the k-means++ converges k-means in 4 iterations, it is happened only two times in 20 runs and in remaining 18 runs either it is equal or more than 5 iterations. The similar results can be observed for remaining data sets in Table 1-3. The results are summarized in the Table 3.

DISCUSSION

Observe the Table 3, in case of Lung Cancer and Serum data sets the SPSS produce results in number of iterations which is less than the number required by k-means++ in its 20 runs. For the remaining data sets k-means++ may reach the solution requiring less number of iterations compared to the SPSS, but it is happened in once or twice in 20 runs.

More over notice closely the last three columns of Table 3, the minimum figures are either equal or close to the SPSS figures and the maximum number are far from the number taken by the SPSS. k-means++ produce different results in different runs. It has to repeat number of times to get good clustering results whereas the SPSS produces single solution in single pass. Observe that the SPSS performs well on Serum, Breast Cancer, Lung Cancer, Ionosphere, Rocks and mines so on, in which dimensions are more than 10(dimensions from 13-60) than Glass, *E. coli*, in which dimensions are less than 10. Therefore, our SPSS algorithm improves its efficiency with increase of dimensions.

As shown in the Table4 and 5, the quality of clusters produced from k-means++ is vary from 2- 10% where as there is no scope for different quality clusters in different runs in SPSS and percentage of quality of clusters from SPSS is nearer to the maximum percentage observed in 20 runs of k-means++.

Experimental results on real data sets (4-60 dimensions, 27-10945 objects and 2-10 clusters) from UCI have demonstrated the effectiveness of the SPSS in producing consistent clustering results.

CONCLUSION

k-means++ is a careful seeding for k-means. However, for good clustering results it has to repeat number of times. The proposed SPSS algorithm is a single pass algorithm yielding unique solution with consistent clustering results compared to k-means++. The SPSS algorithm gives good results when the attributes of the data set are more in number. The computational task required by the SPSS algorithm is less comparative to k-means++ algorithm as the first seed and the minimum probable distance is selected randomly, this may increase the number of iterations and thus it takes more time to reach final solution. Improving the efficiency of the proposed SPSS algorithm for low dimensional data sets and proposing an algorithm to generate number of clusters with optimal centroids is our future endeavor.

REFERENCES

Anderberg, M.R., 1973. Cluster Analysis for Applications. Academic Press, ISBN: 0120576503, pp: 359.

Ankerst, M., M. Breunig, H.P. Kriegel and J. Sander, 1999. OPTICS: Ordering points to identify the clustering structure. Proceeding of ACM SIGMOD International Conference Management of Data Mining, May 31-June 3, ACM Press, Philadelphia, Pennsylvania, United States, pp: 49-60. http://portal.acm.org/citation.cfm?doid=304182.304187

Arthu, D. and S. Vassilvitskii, 2007. k-means++: The advantages of careful seeding. Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis, Jan. 7-9, ACM Press, New Orleans, Louisiana, pp: 1027-1035. http://portal.acm.org/citation.cfm?id=1283494

Astrahan, M.M., 1970. speech analysis by clustering, or the Hyperphoneme method. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0709067

Ball, G.H. and D.J. Hall, 1967. PROMENADE-an online pattern recognition system. Stanford Research Inst. Memo, Stanford University. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0822174

Berkhin, P., 2002. Survey of clustering data mining techniques. Technical Report, Accure Software, SanJose, CA. http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf

Bradley, P.S. and U.M. Fayyad, 1998. Refining initial points for K-means clustering. Proceeding of the 15th International Conference on Machine Learning (ICML'98), July 24-27, ACM Press, Morgan Kaufmann, San Francisco, pp: 91-99. http://portal.acm.org/citation.cfm?id=645527.657466

Aha, D., 1987. UC Irvine machine learning repository. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris

Deelers, S. and S. Auwatanamongkol, 2007. Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. Proc. World Acad. Sci. Eng. Technol., 26: 323-328. http://www.waset.org/pwaset/v26/v26-2.pdf

Dembele, D. and P. Kastner, 2003. Fuzzy C-means method for clustering microarray data. Bioinformatics, 19: 973-980. http://bioinformatics.oxfordjournals.org/cgi/reprint/19/8/973

Duda, R.O. and P.E. Hart, 1973. Pattern Classification and Scene Analysis. John Wiley Sons, New York, ISBN: 0471223611, pp: 482.

Duda, R.O., P.E. Hart and G. David, 2001. Stork Pattern Classification. 2nd Edn., John Wiley and Sons, ISBN: 0471056693, pp: 654.

Eisen, M.B., P.T. Spellman, P.O. Brown and D. Botstein, 1995. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA., 95: 14863-14868. http://www.ncbi.nlm.nih.gov/pubmed/9843981

Ester, M., H. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining, (KDD'96), Germany, pp: 1-6. http://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf

Fahim, A.M., A.M. Salem, F.A. Torkey and M. Ramadan, 2006. An efficient enhanced k-means clustering algorithm. J. Zhejiang Univ. Sci. A., 7: 1626-1633. http://www.zju.edu.cn/jzus/2006/A0610/A061002.pdf

Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, 1996. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, ISBN: 0262560976, pp: 611.

Gersho, A. and R.M. Gray, 1992. Vector Quantization and Signal Compression, Kluwer Academic, Boston, ISBN: 0792391810, pp: 761.

Iyer, V.R., M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee and M.S. Bogosk *et al*., 1999. The transcriptional program in the response of human fibroblast to serum. Science, 283: 283-287. http://www.ncbi.nlm.nih.gov/pubmed/9872747

Jiang, D.J. Pei and A. Zhang, 2005. An interactive approach to mining gene expression data. IEEE. Trans. Knowl. Data Eng., 17: 1363-1380. DOI: 10.1109/TKDE.2005.159

Jain, A.K. and R.C. Dubes, 1988. Algorithms for Clustering Data, Prentice Hall, ISBN: 013022278X, pp: 320.

Katsavounidis, I., C.C.J. Kuo and Z. Zhen, 1994. A new initialization technique for generalized Lloyd iteration. IEEE. Sig. Process. Lett., 1: 144-146. DOI: 10.1109/97.329844

Kaufman, L. and Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, ISBN: 0471878766, pp: 342.

Lloyd, S.P., 1982. Lease square quantization in PCM. IEEE Trans. Inform. Theor., 28: 129-136. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1056489

MacQueen, J.B., 1967. Some Method for Classification and Analysis of Multivariate Observations, Proceeding of the Berkeley Symposium on Mathematical Statistics and Probability, (MSP'67), Berkeley, University of California Press, pp: 281-297. http://projecteuclid.org/DPubS?verb=Display&version=1.0&service=UI&handle=euclid.bsmsp/1200512992&page=record

Pham, D.T., S.S. Dimov and C.D. Nguyen, 2004. Selection of k in K-means clustering. Mech. Eng. Sci., 219: 103-119. http://journals.pepublishing.com/content/pp32548654045644/

Rousseeuw J. and P. Silhouttes, 1987. A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Applied Math., 20: 53-65. DOI: 10.1016/0377-0427(87)90125-7

Tou, J. and R. Gonzales, 1977. Pattern Recognition Principles. Addision-Wesley, Reading, MA., ISBN: 0201075873, pp: 377.

Wu, X., V. Kumar, J.R. Quinlan, J. Ghosh, D.J. Hand and D. Steinberg *et al*., 2008. Top 10 algorithms in data mining. Knowl. Inform. Syst. J., 14: 1-37. http://www.cs.umd.edu/~samir/498/10Algorithms-08.pdf