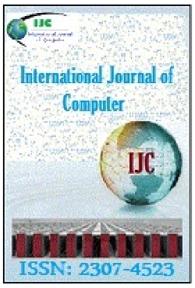
International Journal of Computer (IJC)

ISSN 2307-4531

http://gssrr.org/index.php?journal=InternationalJournalOfComputer&page=index



International Journal of Computer

IJC

ISSN: 2307-4523

Performance Issues on K-Mean Partitioning Clustering Algorithm

Chatti Subbalakshmi [a]*, P.Venkateswara Rao [b], S Krishna Mohan Rao [c]

[a] *Department of Computer Science and Engineering , GNITC, Hyderabad , India*

[b] *Department of Computer Science and Engineering, VNR VJIET, Hyderabad, India*

[a] *subbalakshmichatti@gmail.com*

[b] *pvenkat2004@gmail.com*

[c] *krishnamohan6@yahoo.com*

Abstract

In data mining, cluster analysis is one of challenging field of research. Cluster analysis is called data segmentation. Clustering is process of grouping the data objects such that all objects in same group are similar and object of other group are dissimilar. In literature, many categories of cluster analysis algorithms present. Partitioning methods are one of efficient clustering methods, where data base is partition into groups in iterative relocation procedure. K-means is widely used partition method. In this paper, we presented the k-means algorithm and its mathematical calculations for each step in detailed by taking simple data sets. This will be useful for understanding performance of algorithm. We also executed k-means algorithm with same data set using data mining tool Weka Explorer. The tool displays the final cluster points, but won't give internal steps. In our paper, we present each step calculations and results. This paper helpful to user, who wants know step by step process. We also discuss performance issues of k-means algorithm for further extension.

Keywords: data mining; partition clustering; k-mean.

1. Introduction

Clustering is process of grouping the data objects. Each object in a same group similar to each other and objects in other group are dissimilar.

----

* Corresponding author.
E-mail address: subbalakshmichatti@gmail.com.

There are many clustering techniques are present depends on the way they group the data objects. There many categories of clustering algorithms [1] like Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods and Model-based methods.

One of the categories of clustering methods is partitioning methods. Partitioning method groups given data set *D* of *n* objects or records into *k* number of partitions. Each partition is represented as cluster. The general algorithm for clustering objects in partition methods as: the input parameter given to algorithm are, *D* is data base which to be cluster, *n* is number of objects in data base, *k* is number of clusters or partitions. The output of algorithm is *k* number of partitions or cluster of data objects and $k<=n$. The basic method of partition methods is, partition the data objects into k groups such that

- Each object belongs to exactly on group.
- Each cluster contains at least one object.

The methodology of partition algorithm uses the initial partitioning technique. i.e. initially it construct the k number of partitions and then it uses iterative relocation techniques. To improve performance of the partitioning, by moving objects from one group to other group.

The criterion of partition method for grouping the objects is similarity between the objects. It groups the objects into one partition, where objects are similar or close to each other and objects of other clusters are different. For global optimality, it uses heuristic methods. There are two center based heuristic partitioning clustering algorithms.

1. K-means clustering algorithm [2]: each cluster is represented with mean of partition.
2. K-mediods clustering algorithm: each cluster is represented with one of object of the group which is near to cluster center.

The partition methods works well for small data and medium data sets. The limitations of algorithms are:

1. They works well for small to medium-sized data bases.
2. Well for spherical-shaped clusters.

We can improve the performance of algorithms for large data bases and complex shapes by extending the algorithms.

2. K- MEANS ALGORITHM

The k-means algorithm [2] is centroid- based technique. The k-means algorithm takes n objects from the given data set D and k number of clusters as input parameters. It clusters the objects into k number of groups such that intracluter similarity is maximum and intercluster similarity is minimum. Cluster similarity is measured in mean value of the objects in a cluster which can be called as center of gravity or centroid.

Distance measures: the k-mean algorithm uses distance measures to find then similarity between the objects. It uses most popular distance measure is Euclidian distance, which is defined inequ.1,

$$D\ (\ i, j) = \sqrt{}(x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2 + \ldots\ldots + (x_{in}-x_{jn})^2 \qquad (1)$$

Where $i= (x_{i1}, x_{i2}, \ldots\ldots , x_{in})$ and $j=(y_{j1},y_{j2},\ldots\ldots y_{in})$ are two n-dimensional data objects.

Center of cluster: the k-means algorithm uses mean of the objects in the cluster and mean can be defined in equ.2 as,

$$\text{Mean} = (\ x_{i1} + x_{i2} +\ldots\ldots x_{in}\ ) / n\ ,(\ y_{j1} + y_{j2} +\ldots\ldots + y_{jn}\ ) / n \qquad (2)$$

Where $i=(x_{i1},x_{i2}, \ldots\ldots, x_{in})$ and $j=(y_{j1},y_{j2},\ldots\ldots y_{in})$ are two n-dimensional data objects.

```
Algorithm: $k$-means, algorithm for partitioning given data objects, where each cluster center is represented by
mean value of objects in the cluster.
Inputs:

        1. $k$ : number of clusters
        2. $D$: a data set containing n number of objects.

Output: a set of $k$ clusters.
Procedure:
    Step1: randomly select $k$ number objects from $D$ data set to represent initial cluster centers;

    Step2: for each remaining data objects repeat this step,

        2.1. Calculate distance between data object and each mean of cluster using equ.(1);
        2.2. Compare the distances and assign the data object to cluster mean whose distance is minimum
             than others;

    Step3: update the new mean of each cluster to represent new cluster center using equ.(2);

    Step4: go to step2 until there is no change in clusters;
```

2.1 Illustration of algorithm

In this section, we present mathematical calculation for each step of algorithm while it is executing. In order to show the illustration of k-means partition clustering algorithm[3], we took random the two-dimensional data set D with eight points as (x, y) representing the location as, *D* = { (2,10), (2,5), (8,4), (5,8), (7,5), (6,4), (1,2), (4,9) }

For finding the distance between the data point and mean of cluster, we use the Euclidean distance function as defined by *k*-means algorithm. We executed the algorithm by taking the number of clusters, $k = 3$. At the beginning of algorithm, initially we assigned (2, 10), (5, 8), (1, 2) as center of each cluster, respectively. Here, we have shown steps in execution of algorithm in detailed.

The Input parameters given by user are:

Let us consider the data sets, *D* = { (2,10), (2,5), (8,4), (5,8), (7,5), (6,4), (1,2), (4,9) } and

The number of clusters, $k = 3$

First Iteration of algorithm:

In first iteration, initial cluster centers are, c1= (2, 10) , c2 = (5, 8) and c3 = (1, 2)

Step.1: Pick the each data object or point represented as P(x, y) from *D* and calculate the distance between object and cluster centers using equ.1.

Let us consider the fist point P(2,10) and mean of cluster center -1 is M1(2,10)

The Euclidean distance between point P(2,10) and mean of cluster-1 M1 (2,10) is ,

$PM1=\sqrt{(2-2)^2 + (10-10)^2} = \sqrt{0^2+0^2} = \sqrt{0+0} = \sqrt{0} =0$

The Euclidean distance between point P(2,10) and mean of cluster-2 is M2(5, 8) is ,

$PM2= \sqrt{(2-5)^2 + (10 – 8)^2} = \sqrt{(-3)^2 + (2)^2} = \sqrt{9+4} = \sqrt{13} = 3.60$

The Euclidean distance between point P(2,10) and mean of cluster-3 is M 3(1, 2) is ,

$PM3= \sqrt{(2 – 1)^2 + (10 – 2)^2} = \sqrt{12 +82} = \sqrt{1+64} = \sqrt{65} = 8.062258$

Step. 2: Compare the distances and assign the object or point to the cluster which is similar to object based on distance.
The calculated distances from the above step are, PM1=0 , PM2= 3.60 and PM3= 8.062258 and PM1<PM2<PM3.

Therefore the point is near to cluster-1 mean than the other cluster mean. Assign the object (2,10) to cluster-1.

Same way we calculated distances, compare the distances and assigned the objects to which the distance is minimum. We have shown this process results in the table.1 given below.

Table 1. Results of first iteration of k-means algorithm

data point (P)	Euclidian Distance=PM1	Euclidian Distance=PM2	Euclidian Distance=PM3	Minimum distance	(re)Assigned cluster
(2,10)	0	3.60	8.062258	0	C1
(2, 5)	4.242641	5	3.162278	3.162278	C3
(8,4)	8.485281	5	7.28011	0	C2
(5,8)	3.605551	0	7.211103	0	C2
(7,5)	7.071068	3.605551	6.708204	3.605551	C2
(6,4)	7.28011	4.123106	5.385165	4.123106	C2
(1,2)	8.062258	7.211103	0	0	C3
(4,9)	2.236068	1.414214	7.615773	1.414214	C2

After completion of first iteration of algorithm, data points (re)assigned to clusters and their cluster means are given in the table 2.

Table 2. After execution of first iteration, the cluster points

	Cluster1	Cluster2	Cluster3
	(2,10)	(8,9)	(2,5)
		(5,8)	(1,2)
		(7,5)	
		(4,9)	
		(6,9)	
Cluster mean	(2,10)	(6,6)	(1.5,3.5)

Second Iteration:

From the first iteration, the cluster centers are M1(2,10), M2(6, 6) and M3(1.5, 3.5) given for next iteration and same way of first iteration we calculated distance between the data point and new cluster mean. We reassigned the data points to cluster means which is similar to the data point as like in first iteration. The calculations and results are given in the table 3. And resultant cluster points are given in the table 4.

Table 3. Result of second iteration

data point (P)	Euclidian Distance=PM1	Euclidian Distance=PM2	Euclidian Distance=PM3	Minimum distance	(re)Assigned cluster
(2,10)	0			0	C1
(2, 5)	5	4.123106	1.581139	1.581139	C3
(8,4)	7.211103	2.828427	6.519202	2.828427	C2
(5,8)	3.605551	2.236068	5.700877	2.236068	C2
(7,5)	7.071068	1.414214	5.700877	1.414214	C2
(6,4)	7.211103	2	4.527693	2	C2
(1,2)	8.062258	6.403124	0.707107	0.707107	C3
(4,9)	2.236068	3.605551	6.041523	2.236068	C1

Table 4. After execution of second iteration, reassigned cluster points

	Cluster1	Cluster2	Cluster3
	(2,10) (4, 9)	(8,9) (5,8) (7,5) (6,4)	(2,5) (1,2)
Cluster mean	(3, 9.5)	(6.5, 5.25)	(1.5,3.5)

Third Iteration:

From the second iteration, the cluster centers are M1(3, 9.5), M2(6.5, 5.25) and M3(1.5, 3.5) given for next iteration and same way of first iteration we calculated distance between the data point and new cluster mean. We reassigned the data points to cluster means which is similar to the data point as like in first iteration. The calculations and results are given in the table 5. And resultant cluster points are given in the table 6.

Table 5. Result of third iteration

data point (P)	Euclidian Distance=PM1	Euclidian Distance=PM2	Euclidian Distance=PM3	Minimum distance	(re)Assigned cluster
(2,10)	1.118034	6.542935	6.519202	1.118034	C1
(2, 5)	4.609772	4.506939	1.581139	1.581139	C3
(8,4)	7.433034	1.783956	6.670832	1.783956	C2
(5,8)	2.5	3.132491	5.700877	2.5	C1

(7,5)	6.020797	1.520691	5.700877	1.520691	C2
(6,4)	6.264982	1.346291	4.527693	1.346291	C2
(1,2)	7.762087	6.388271	1.581139	1.581139	C3
(4,9)	1.118034	4.930771	6.519202	1.118034	C1

Table 4. After execution of third iteration, cluster points

	Cluster1	Cluster2	Cluster3
	(2,10)	(8,4)	(2,5)
	(5, 8)	(7,5)	(1,2)
	(4, 9)	(6, 4)	
Cluster mean	(3.66, 9)	(7, 4.33)	(1.5,3.5)

Fourth Iteration:

From the third iteration, the cluster centers are M1 (3.66, 9), M2 (7, 4.33) and M3 (1.5, 3.5) given for next iteration and same way of first iteration we calculated distance between the data point and new cluster mean. We reassigned the data points to cluster means which is similar to the data point as like in first iteration. The calculations and results are given in the table 7. And resultant cluster points are given in the table 8.

Table7. Result of fourth iteration

data point (P)	Euclidian Distance=PM1	Euclidian Distance=PM2	Euclidian Distance=PM3	Minimum distance	(re)Assigned cluster
(2,10)	1.9364917	7.5596891	6.5192024		C1
(2, 5)	4.3307736	5.0446903	1.5811388		C3
(8,4)	6.6208459	1.0530432	6.6208459		C2
(5,8)	1.6720048	4.1795813	5.7008771		C1
(7,5)	5.2111035	1.33	5.7008771		C2
(6,4)	5.520471	1.0530432	4.5276926		C2
(1,2)	7.4883643	6.4257295	1.5811388		C3
(4,9)	1.66	5.5505765	6.041523		C1

Table 4. After execution of third iteration, cluster points

	Cluster1	Cluster2	Cluster3
	(2,10)	(8,4)	(2,5)
	(5, 8)	(7,5)	(1,2)
	(4, 9)	(6, 4)	
Cluster mean	(3.66, 9)	(7, 4.33)	(1.5,3.5)

Termination condition: as per k-means algorithm we can terminate the process until there are no change cluster points. The algorithm is terminated after completion of third iteration. The number of iterations *t* always less than or equal to number of data points in data set. i.e. $t \leq$ n. another termination condition is, the process is terminates until the criterion function convergence. It uses Square error criterion [4] and defined in Equ. 3 as,

$$E = \sum_i \sum_{p \in Ci} | p - m_i |^2 \quad (3)$$

Where $i= 1..k$; E is the sum of the square error for all objects in the data set; $\boldsymbol{p}$ is point in space representing a given object; and $\boldsymbol{m_i}$ is the mean of cluster **C***i*.

In fourth iteration there is no change in cluster points, which indicates the termination of algorithm. After the execution of k-means partition clustering algorithm on given data set *D*, the clusters objects are:

Cluster-1: (2, 10), (5, 8), (4, 9)

Cluster-2: (8, 4), (7, 5), (6, 4)

Cluster-3: (2, 5), (1, 2)

3. Experimental results

We experiment the k-means algorithm using Weka GUI Explorer on different data sets. Weka is open source data mining tool developed in Java, we can perform the many data mining task. It is providing GUI environment for data mining process.

a. Experiment on small data set

We verified our results by executing k-means algorithm with same data set using the Weka software. Here we are giving steps for execution algorithm in Weka:

Step1. Create ARFF file for data set given in table 7.

Table. 7. ARFF file for data set

```
@relation sample

@attribute x numeric
@attribute y numeric
 @data
2,10,5,8,4,9,8,4,7,5,6,4,2,5,1,2
```

1. Open this file in Weka Explorer pre-process window.
2. Select SimpleKmeans clustering algorithm from cluster choose option and set number of cluster as 3.
3. Choose cluster mode as training data set.
4. Click on start.

It displays the run information of algorithm given here:

```
=== Run information ===
Scheme:weka.clusterers.SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -O -S 10
Relation:     sample
Instances:    8
Attributes:   2
          x
          y
Test mode:evaluate on training data
=== Model and evaluation on training set ===
kMeans
======
Number of iterations: 4
Within cluster sum of squared errors: 0.2582376700680272
Missing values globally replaced with mean/mode
Cluster centroids:
                 Cluster#
Attribute    Full Data       0        1           2
                (8)        (2)      (3)         (3)
=========================================
x            4.375        1.5    3.6667        7
y            5.875        3.5       9         4.3333

Time taken to build model (full training data) : 0 seconds
 === Model and evaluation on training set ===
 Clustered Instances
  0     2 ( 25%)
  1     3 ( 38%)
  2     3 ( 38%)
```

The final result is same as our illustration of the algorithm. We have given the every iteration of the algorithm results, where software is giving the final clustering points as output.

b. Performance issues

We tested the k-means algorithm in synthetic data sets and identified the advantages and limitations of algorithms. In literature there many extended k-means algorithms are present to improve the performance [6][10]. The merits of k-means algorithm:

- k-means algorithm works well for compact clouds.
- It is scalable and efficient in processing large data sets.
- This method terminates often at a local optimum.
- The computational complexity is O(nkt)[7], where n is number of data objects, k is number of clusters and t is number of iterations.
- The number of iterations [6] always less than or equal to n. i.e. k<=n and t<=n.

The main drawbacks of k-means algorithm are:

1. Number of clusters

- We need give the number of clusters before execution of algorithm as input which is fixed.
- Deciding the number of clusters before is not supportable for all type applications.
- In dynamic environment where data set changes over time, we can not determine the number of cluster to be group in prior.
- It is suitable only on static data set where there is no change.

2. Initial cluster centers

Basically it is centroid-based clustering algorithm, where we need to select the starting cluster centers before start of algorithm. For that we have to select randomly k number of clusters to represent k cluster centers which may influence the performance of algorithm.

3. Shape of cluster

The k-means algorithm is not suitable for non-convex shapes or clusters of very different size [9].

4. Mean of cluster

The k-means algorithm can be applied only when the mean of a cluster is defined. But this may not be the suitable in some applications, such as when data with categorical attributes are present.

5. Noise and outlier

The k-means algorithm is not able to handle noise and outlier data points because a small number of such data can be substantially influence the mean value [8].

4. Conclusion and Feature work

In this paper, we presented the general method of partition clustering method and also mention the merits and demerits of partition clustering algorithms. We presented mathematical calculations of k-means algorithm for each iteration. We executed k-means algorithm using Weka data mining tool. Finally we presented the performance issues of k-means algorithm, which can be helpful for extension of k-means algorithm. Mainly our paper is useful for beginners to know the step by step process of k-means algorithm and its merits and demerits.

References

[1]. J.A. Hartigan (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

[2]. Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". *Journal of the Royal Statistical Society, Series C* **28** (1): 100–108. JSTOR 2346830

[3]. Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002). "An efficient k-means clustering algorithm: Analysis and implementation". *IEEE Trans. Pattern Analysis and Machine Intelligence* **24**: 881–892. doi:10.1109/TPAMI.2002.1017616. Retrieved 2009-04-24.

[4]. Lloyd, S. P. (1957). "Least square quantization in PCM". *Bell Telephone Laboratories Paper*. Published in journal much later: Lloyd., S. P. (1982). "Least squares quantization in PCM". *IEEE Transactions on Information Theory* **28** (2): 129–137.doi:10.1109/TIT.1982.1056489. Retrieved 2009-04-15

[5]. Hamerly, G. and Elkan, C. (2002). "Alternatives to the k-means algorithm that find better clusterings".*Proceedings of the eleventh international conference on Information and knowledge management (CIKM).*

[6]. Vattani., A. (2011). "k-means requires exponentially many iterations even in the plane". *Discrete and Computational Geometry* **45** (4): 596–616.doi:10.1007/s00454-011-9340-1.

[7]. Arthur, D.; Manthey, B.; Roeglin, H. (2009). "k-means has polynomial smoothed complexity". *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS).*

[8]. Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NP-hardness of Euclidean sum-of-squares clustering".*Machine Learning* **75**: 245–249. doi:10.1007/s10994-009-5103-0.

[9]. Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. (2009). "The Planar k-Means Problem is NP-Hard". *Lecture Notes in Computer Science* **5431**: 274–285. doi:10.1007/978-3-642-00202-1_24.

[10]. Arthur; Abhishek Bhowmick (2009). *A theoretical analysis of Lloyd's algorithm for k-means clustering*(Thesis).