On Initial Effects of the k-Means Clustering

Sherri Burks, Greg Harrell, and Jin Wang
Department of Mathematics and Computer Science
Valdosta State University, Valdosta, Georgia 31698 USA

Abstract - *There are many research studies conducted in order to find a more optimal way to initialize the k-means algorithm, also referred to as Lloyd's algorithm. Despite the appreciated efficiency of the k-means process, occasionally it may return a less than optimal clustering solution. It is widely believed that modifications to the initialization process will improve results. Here, the choice of initial centroids for the k-means clustering technique is reviewed with respect to efficiency in stabilizing or convergence with different initialization methods. Several proposed initialization techniques are evaluated on a two dimensional model in an attempt to verify or reproduce results similar to those of the studies chosen.*

Keywords: Algorithm; Cluster; Data Mining; K-Means

1 Introduction

1.1 Definition

The k-means clustering algorithm is a procedure that is widely applied in data mining, biometrics, and signals processing to aid in aggregating or visualizing related data. It is used to establish a set number (k) partitions or clusters of a dataset in which each element of the cluster is most like, or within the closest proximity of that cluster's mean. This is determined by a series of recursive calls that begins with assigning a data point to one of the k arbitrarily chosen initial cluster centroids based upon its closest Euclidian proximity, defined below in Figure 1.

$$d_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

Once all the data points have been assigned to a cluster, the cluster means are computed and the data points are re-assigned to the new centroids. The process is repeated until it converges and no changes in clusters or centroids are observed. Section 2 explains the k-means algorithm more simply.

The effects of the initialization of the k-means process is claimed to have a significant impact upon the stability and optimization of the results. The deliberations surrounding initializing the event consists of determining the quantity of clusters (k) and thus the number of centroids, which at times may be determined by the user's application, and also consists of the method used in choosing the initial centroids for the first iteration. Although some of the methods reviewed address determining the proper quantity of clusters, the scope of this study will focus on the latter. Several methods of initializing the k-means algorithm will be reviewed and the effects of the initialization compared using the same data sets and number of clusters.

1.2 Application

As the amount of large scale data collection continues to increase, the need for efficient and reliable methods to process big data becomes essential. The utilization of the k-means algorithm as a solution to clustering problems can be found in applications from data mining to fuzzy vectors and visual graphics. Implementations of the k-means algorithm are extensively found in data visualization for purposes of data mining and recommender systems. These recommender systems may be used to find venues or items for individuals of like mindedness, or location based similarity such as a nearest neighbor. Additionally, variations of the k-means are found in decision engines, medical imaging, routing, and in signal cleaning, removing noise and detecting outliers. Determining a better initialization method may result in a faster convergence or may avoid a less than optimal solution. More reliable clustering results would help facilitate trust and reliance upon machine learning based systems, particularly in those of a critical nature such as in medical applications.

Traditional k-means is a greedy algorithm that can present issues with stability, particularly with the selection of outliers and low density areas as cluster centers. For example, a situation may be observed where an element may be equally distant between two centroids and may oscillate between two clusters in assignment. Additionally, k-means may exhibit a tendency of convergence to a local minima. It does not ensure a global minima, nor does it guarantee unique clustering [1]. It has been reported that noise or isolated outliers may greatly affect average values [4] and skew cluster regionalization. Finding a more advantageous initialization method for k-means should produce more optimal results, faster convergence and stability, and may improve the user's trust and reliability of the results in data analysis applications.

2 K-means algorithm

2.1 Algorithm pseudo-code

The basic algorithm used for k-means is as follows:

```
Input:
        S:  set of data points, ex.: (x, y)
        k: the number of clusters

Output:
        C:  set of k clusters, indicated by Ci …Ck
Process:
        1.  Choose k points randomly from S as the initial
            centroids (means or center of mass)
        2.  Calculate the Euclidian distance from each point to
            each centroid.  Assign the point to the set of the
            nearest cluster Ci.
        3.  Recalculate the means for each cluster and shift the
            centroid as necessary.
        4.  Repeat step 2 and 3 until there are no changes.
```

2.2 Computer implementation

A java program was executed on a 64-bit Intel Core i5-3337U CPU @ 1.80GHz with 4 GB memory running the Windows 8.1 Operating System. The original k-means clustering was implemented by generating random data sets of two dimensional Cartesian coordinate values. As each coordinate was created it was checked to make sure it was not a duplicate before being added to the data set. This same data set is used in later comparisons for all initialization process types. A pre-determined number of clusters was established to be 3. (Future comparison studies may involve 4 and 5 clusters on larger data sets). A random centroid was chosen to be each cluster's initial center, and the k-means algorithm proceeded by assigning points to nearby cluster centers and then adjusting the location of the centroid based upon the mean value of each cluster. This execution continued until changes in cluster assignment or movements of the centroids were no longer observed and the process stabilized. In order to accomplish this, the sets of coordinates and centroids were stored as linked lists of point objects and Boolean variables were used to identify when convergence was reached. New datasets are generated for subsequent comparisons, and results are averaged.

3 K-means initial cluster center

3.1 Initialization Effects

Although deciding upon the best manner to initialize the cluster center is of some contention, there seems to be much agreement on the effects of poor initialization and the importance of choosing a proper method of seeding the process. There is concurrence that the choice of the initial cluster centers may result in different clustering results of the same data set. In addition, the number of iterations that the process requires before stabilizing may be affected by a poor initialization. Concerns of becoming stuck in a local minimum and instability issues as defined by [4] as the random sampled mean minimal matching distance between two clusters on two sets of data points. According to [6], the initialization effects are more appreciated when there are large numbers of clusters.

3.2 K-means sensitivity (stability) analysis

Different choices of initialized cluster centers may lead to unstable clustering results as it may likely produce different partitions. [5] Such instability is described by [4] as the random sampled mean minimal matching distance between two clusters on two sets of data points. Noise, outlying data points, and isolated data points affect the mean values of dense clusters, and tend to bias the clustering into less than optimal results. In one instance that was observed on a sample run, the partitions were not the best choice, as the cluster that had two extreme outliers stretched over a greater margin on the y-axis, rather than having some of those points assigned to a more logical cluster. In this case, the centers were closer together and less than optimal to start. Another observation was made where the partitioning took place in a more horizontal fashion when a slightly more vertical partitioning appeared more likely.

Numerous studies have been published proposing solutions to these issues, recommending various methods of the initialization process of selecting the first set of cluster centroids. Some of the less involved approaches that use comparable methods will be addressed in section 4.

4 K-means initial selection algorithms

There are a number of sources of initial selection algorithms to choose from. Some of the studies reviewed focused on discovering the proper number of clusters prior to running the clustering process. One publication additionally noted the importance of the number of centroids with respect to the number of clusters. Because of this, techniques with a common, pre-determined number of clusters (k) were used in this study. However all of the techniques reviewed emphasized the importance of finding the appropriate centroids to begin the clustering process. The options reviewed ranged as follows: using weights based upon probability, choosing a centroid location based upon density, subdividing clusters into smaller subsections prior to choosing a centroid, using a graph based method, and combinations of these techniques.

4.1 Survey of algorithms

4.1.1 K-means++

The k-means++ algorithm uses an initialization method of choosing the random starting centroids based upon the probability of that point being a center in an attempt to produce clusters that are more diffuse. The first center is chosen at random. Each additional k-1 centers are chosen based upon a weighted probability such that once the first random centroid is set the selection of each successive center is affected by the other centers that have been chosen prior. Once all of the centers are chosen, the process proceeds based upon the original k-means algorithm, assigning points to the clusters until a convergence is reached. The calculation of the probability for the k-1 centers is a ratio of a candidate center point's distance to the other centers. It is measured as the squared shortest distance of the point to the closest centroid divided by the sum of the squared distance of all other candidate points to the center. One criticism of this technique is that it is not scalable for big data. [3]

4.1.2 Density

A density based approach to cluster initialization relies heavily upon accurately determining those areas of high density. Density based methods do not rely upon random selection, but instead determine the density of an area within a radius and then find other high density areas that are further apart. One approach selects the initial centroid from the highest density area. That centroid and its assigned cluster are removed from the data pool and the process repeats to find the next centroid and its corresponding cluster, continuing until all k clusters have been determined. Here a strong emphasis must be placed upon establishing a proper radius. Incorporating distance into the decision is more harmonious with the source distribution [5] and in the absence of outliers it yields better clustering results by having the centroids in those high density areas farther from each other. *It is of great importance to note that this procedure assumes that the low density areas may be outliers.* Any density based approach levies great importance upon deciding what constitutes an area of high density.

4.1.3 Reverse nearest neighbor and coupling

[1] This concept combines the method of conducting a proximity search for a reversed nearest neighbor and the method of determining and eliminating candidate centroids of similar likeness. A reverse nearest neighbor is the point(s) to which the current point is closest to, rather than the closest neighbor to that point. This approach transpires by creating three sets. To begin, all points are included in the candidate set. Representative points are selected by finding and counting the reversed nearest neighbor for each point in the candidate set, and adding the point with the highest count to the representative set. The point and its corresponding points are removed from the candidate set. This repeats until there are only points with a count of one remaining. Choosing the centroids are selected from this representative set using a coupling technique. The average distance between all points is calculated and the coupling degree between all pairs of points is calculated. Points with the largest count of neighbors and coupling degree are selected. A mean point is chosen as calculated for that point and its neighbors, and added to the centroid set. The point and its corresponding neighbors are removed from the representative set and the process repeats until all k initial cluster centroids have been selected. Then the k-means process continues as usual.

4.1.4 Mean-shift and k-means++

A multi-faceted approach from [6] involves combining methods used in mean shift and k-means++. As a premise, it requires reducing the number of dimensions down to two dimensional data consisting of the two most significant, independent variables to represent the overall data. The first is chosen from the highest calculated coefficient of variation and the second variable is the one with the lowest calculated correlation coefficient. (Test data for this study is already two dimensional. Therefore this process is eliminated in test comparisons in order to both keep the evaluation on a more even standing and tighten the focus on cluster initialization centroid selection effects.) This is followed by determining an appropriate radius for the likeness neighborhood, essentially determining the cluster boundaries or areas of most data similarity. Finally the cluster centers are chosen. The first is an arbitrarily chosen centroid. The mean of the data points within that centroid's radius is calculated and the centroid is shifted to the nearest data point. A probability ratio is calculated similar to that of k-means++ where the shortest distance of each data point to its corresponding centroid is divided by the maximum distance to from all points to the centroids. The calculated ratio is the likeliness of that point being a candidate centroid and is used in selecting the remaining centroids. Each new centroid that is selected is shifted towards the cluster mean, and the process continues until all centroids and clusters have been established.

4.2 K-means algorithm with different initial selections pseudo-code

4.2.1 K-means++

The k-means++ algorithm is as follows:

```
Input:
        S:  set of data points, ex.: (x, y)
        k: the number of clusters

Output:
        C:  set of k clusters, indicated by Ci …Ck

Process:
        1.  Choose a point randomly from S as the first initial
            centroid C1.

        2.  Choose the following k-1 centroids C2… Ck by
            determining probability in the following manner:
            For all remaining points, find $D(x)^2$:  Square the
            Euclidian distance from a candidate point to C1.
            Add these values to an accumulator that holds the
            sum of $D(xi…k-1)^2$ distances for all candidate
            points. Divide the $D(x)^2$ for each point by that
            accumulated  sum to obtain that point's
            probability. Finally choose the centroid randomly
            from the candidates based upon the weighted
            probabilities.
```

3. Repeat step 2 until all k centroids have been chosen.

4. Once all initial centroids have been established, continue by using the original k-means process of assigning the data points to cluster centers and re-calculating the centroids until the process stabilizes.

4.2.2 Density

The density based algorithm discussed is as follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters
r: the radius used for calculating density

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

1. For each point, count the number of nearest neighbors within a certain radius.

2. For the first centroid, select the point with the highest neighbor count and add it to set C.

3. Remove that point and its nearest neighbors from the set of available points to pick from.

4. Repeat step 2 and 3 until all k centroids have been chosen.

5. Proceed with clustering using the original k-means process of assigning the data points to cluster centers and re-calculating the centroids until the process stabilizes.

4.2.3 Reverse nearest neighbor and coupling

The reverse nearest neighbor and coupling algorithm follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters
r: the radius used for calculating density

Output:

C: set of k clusters, indicated by Ci …Ck

Additional structures used:

CS: set of candidate points
RNN: a list of each point's RNN count
RS: representative set

Process:

Phase 1:

1. All points in S are included in CS.
2. For each point, its reverse nearest neighbors (RNN) are counted and the number is stored in the RNN list.
3. The point with the max RNN count is added to the RS set and deleted from the CS set. Its corresponding RNN points are also deleted.
4. Repeat step 3 until the only values remaining in RNN equals one.

Phase 2:

5. Calculate the average distance between all points.
6. Calculate the coupling degree between all possible pairs of points in the RS set.
7. Select the point with the greatest number of neighbors and coupling degree and calculate the mean for that point and its neighbors.
8. Add that point to the set of cluster centroids, C. Delete that point and its neighbors from RS.
9. Repeat steps 7 – 8 until k centroids have been selected.

4.2.4 Mean-shift and k-means++

This method proceeds in the following manner:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

Phase 1:

1. Reduce the number of dimensions down to a two dimensional representative subspace by selecting the two most significant, independent variables. The first is chosen from the highest calculated coefficient of variation and the second variable is the one with the lowest calculated correlation coefficient. (Assume this has been completed for the study comparison).

Phase 2:

2. Determine the radius:
   a) Randomly select 100 points from S.
   b) Compute each point's distance to its nearest neighbor.
   c) The radius equals 4 times the max of the distances found in step b.

Phase 3:

2. Randomly choose a point from S and find the mean of its neighbors within the radius from 2.c.

3. Shift the centroid to the point nearest the calculated mean. Repeat this until the centroid location stabilizes.

4. Use a probability ratio to find the next candidate centroid:
   a) Find the minimum distance from each data point in S to the centers.
   b) Divide that value by maximum distance from all data to the centers.
   c) Use that ratio as the probability for the point to be a centroid.
   d) Choose the most probable point as the next centroid.

6. Shift the cluster center as in step 4 until convergence is reached.

7. Repeat steps 5 -6 until all k cluster centroids are found.

4.3 Computer implementation – programming

A java program was implemented on a 64-bit Intel Core i5-3337U CPU @ 1.80GHz with 4 GB memory running the Windows 8.1 Operating System. Random sets of two dimensional Cartesian coordinates were generated. For simplicity and scientific fidelity for comparisons the same data sets and the same number of clusters were used for all initializing algorithms. Array lists were used to store coordinates (as point objects) and centroids (also point objects). A table was used to keep track of the assignment of points to centers. Also, a timer was placed to keep track of execution time of the entire process from initialization to stabilization so that the expense of a complex seeding can be measured against a simpler initialization process. Additionally, a counter was used to keep track of the number of iterations for the k-means process to stabilize (post-initialization) so that the effectiveness and efficiency of the particular methods may be evaluated. This is in line with experiments conducted in study [1], asserting that lower iterations to achieve convergence indicates a more accurate initial centroid selection and the execution time of the algorithm can be used as a measure of performance. For the initialization procedures that required a radius, the same method was used in determining the cluster radius as outlined in [6]. The reasoning for this is twofold. It asserts that the techniques are compared in a more even manner and compensates for those studies using radius with an unspecified definition. Finally, distances from cluster points to their centroid were stored for comparing the overall partitioning outcome for each technique.

5 Comparisons

In the tables below, each of the methods were performed concurrently on the same data set. Often each process returned different resulting centroids, but in general, they were in fairly close proximity to the other methods centroids. The graphical placement was not studied in detail due to time constraints. However, the number of iterations and elapsed time indicate the efficiency of the initialization technique as the measurements were taken after the respective initialization process was conducted. It measures how much elapsed time the k-means process required to reach a convergence based upon how it was initialized and how many iterations were completed. In order to calculate the time, the system clock was started at the beginning of the k-means clustering and stopped at the point of convergence. For the comparisons, the test was repeated on newly generated datasets and the iterations and elapsed time was averaged for each process. As the datasets size increased, the initialization time required increased dramatically while the time for the k-means to converge remained relatively the same. The time expense of the density based approach and the combined approaches were quite obvious on large amounts of data. Future comparisons will include a clocked estimate for the overall process in order to compare the overall expense of a complex, highly iterative initialization.

Table 1. Results of Dataset 1 of 5000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	4.2	21.8
K-means++	5.2	21.2
Density	4.3	13.9
MeanShift & K-means++	4.3	19

Table 2. Results of Dataset 2 of 10000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	4.4	35.9
K-means++	5.1	35.9
Density	5.9	43.8
MeanShift & K-means++	4.4	28.2

Table 3. Results of Dataset 3 of 15000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	7.5	81.2
K-means++	6.7	69
Density	6.9	70
MeanShift & K-means++	6.2	61.3

Table 4. Results of Dataset 4 of 20000 points with k =4

Method	Iterations	Elapsed Time (ms)
K-means	6.4	115.7
K-means++	7.9	142.2
Density	6.7	118.9
MeanShift & K-means++	5.6	101.5

6 Summary and conclusions

There appears to be universal agreement among the papers reviewed that the basic k-means process is greatly affected by the initialization of the centroids and number of clusters and may yield a less than optimal solution with a localized minima. [1,2,3,4,5,6] The number of clusters has significant importance but it was deemed important enough for a separate study independent from the centroid initialization, much like in study [4]. Many studies examined evaluated attempts at modifying both aspects. However, it was decided that examining the two factors separately will yield a better understanding of the impact of each component. The focus of this paper centralized on methods of seeding the cluster centers.

K-means++ initialization allows for a bias that an initial centroid will be further away from the other selected centers, resulting in improved clustering and less iterations to reach convergence [6][2]. In addition, there is also the possibility that the initial cluster chosen randomly, C1, could itself be an outlier, thus affecting the distance calculations and adding iterations prior to k-means stabilization.

Reverse nearest neighbor and coupling is expensive and requires several iterations and distance calculations though sets of points. The concept seems solid, however attempts to replicate the procedures used by [1] were difficult, as the method for selecting the point with the max neighbor count and coupling degree was not explicitly specified and therefore left to interpretation as to whether the values should be combined or a ratio, or which value took precedence. As a result experiments using this technique were incomplete and omitted from the results comparison charts.

Mean shift with k-means++ does not suffer the same vulnerability of the k-means++ and performed well in testing. Even though the initial center is randomly chosen, the focal point is shifted prior to determining the probability ratio and choosing the next centroid, thus reducing the effects of the first randomly chosen point occurring in a low density area. The algorithm suggested by [6] did not clearly indicate how to prevent the selection of a duplicate point as a candidate. Therefore, that was subject to programming discern. When the process continues according to the algorithm suggested, the k-means shuffles the centroids, eliminating the problem of the duplicate. However, the results of the study are not reliable because of the potential duplicates that incur despite the coding corrections. The mean shift did not facilitate for the overlap of radii between clusters.

With additional time on this study, more measurable and quantifiable calculations would be presented to determine the accuracy of each process in finding an optimal cluster center and additional initialization processes would be added for testing. In addition, more visual, graphical representations would be added. It did seem that some of the initialization processes were more costly than the amount of iterations saved on the k-means process. This can be determined with the addition of a timestamp prior to beginning each process and at the finish of the k-means.

Even the crude experimentation in this study made it apparent that the initialization of the k-means process unquestionably affects the resulting centroids and the assigned cluster of data points. Implementing comparisons of the distance between resulting centroids from each process and the size (count of points) of the clusters produced would likely reinforce this conclusion and further prove which initialization techniques produce more accurate clustering results.

In the future it would be interesting to perform a more exhaustive study on initialization effects by conducting a stripped k-means clustering comparison after hand-choosing the centers and comparing them with random on some pre-determined data sets such as the Iris data set used in study [1]. First by specifically choosing points that are extremities and comparing them to all of the different initializations mentioned here, then by specifically choosing centroids that are located in less dense areas, and finally choosing centroids in the 'optimal' tight, dense areas. Illustrating the worst case scenarios then comparing the outcomes may lead to new ideas or combinations of cluster seeding such as pre-processing via noise filtering prior to clustering.

7 References

[1] Ahmed, A. and Ashour, W. "An Initialization Method for the K-means Algorithm using RNN and Coupling Degree." International Journal of Computer Applications (0795-8887) Volume 25- No.1. (July 2011).

[2] Arthur, D. and Vassilvitskii, S. "K-Means++: The Advantages of Careful Seeding." Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics (Philadelphia, USA, 2007). 1027-1035.

[3] Bahman, B., Moseley, B., Kumar, R. Vattani, A., and Vassilivitskli, S. "Scalable K-Means++". Proceedings of the VLDB Endowment. Volume 5, Number 7. (Istanbul, Turkey, 2012), 622-633.

[4] Bubeck, S, Melia, M., and Luxburg, U. "How The Initialization Affects The Stability of the K-Means Algorithm". EDP Sciences. SMAI (2012). PS 16 436-452. http://www.esaun-ps.org.

[5] Joshi, K. and Nalwade, P. "Modified K-Means for Better Initial Cluster Centres". International Journal of Computer Science and Mobile Computing. Volume 2, Issue 7 (July 2013) 219-223.

[6] Qiao, J. and Lu, Y. "A New Algorithm for Choosing Cluster Centers for K-Means". Proceedings of 2nd International Conference on Computer Science and Electronics Engineering (Paris, France, 2013). Atlantic Press. 0527-0530.