Homework 1
Applied Machine Learning
Fall 2017
CSCI-P 556

Ashay Sawant
ahsawant@umail.iu.edu

September 18, 2017

"All the work herein is solely mine."

Problem 1

From textbook, Chapter 10 exercise 3 (Page 414).
a) Plot the following observations.

Observations	X_1	X_2
1	1	4
2	1	3
3	0	4
4	5	1
5	6	2
6	4	0

Answer:

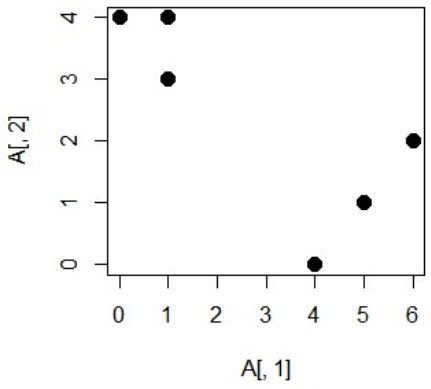


Figure 1: Plot of 6 data points

b) Randomly assign a cluster label to each observation. You can use the sample() command in R to do this. Report the cluster labels for each observation.

Answer:

Observations	X_1	X_2	Label
1	1	4	1
2	1	3	2
3	0	4	1
4	5	1	2
5	6	2	2
6	4	0	1

c) Compute the centroid for each cluster. Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation. Repeat (c) and (d) until the answers obtained stop changing.

Answer:
Iteration 1 :
Centroid 1 - (1.666667, 2.666667) Centroid 2 - (4, 2)
Iteration 2
Centroid 1 - (0.6666667, 3.666667) Centroid 2 - (5, 1)
Iteration 3
Centroid 1 - (0.6666667, 3.666667) Centroid 2 - (5, 1)

Since, there were no changes in centroid in Iteration 2 and 3. Hence, K-means converges at this momment.

**** Final Centroids ****
Centroid 1 - (0.6666667, 3.666667) Centroid 2 - (5, 1)

f) In your plot from (a), color the observations according to the cluster labels obtained.
Answer:

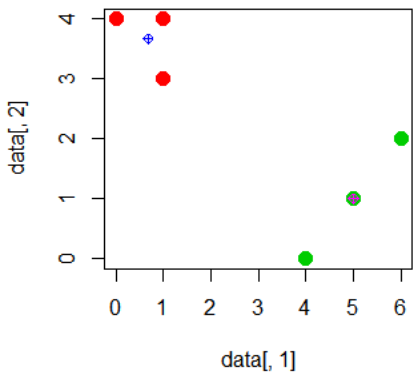


Figure 2: Plot of data points in 2 cluster

Problem 2 [20 points]

The pseudo-code for k-means and a running example of k-means on a small data set are provided above. Answer the following questions

2.1 Does k-means always converge? Given your answer, a bound on the iterate must be included. How is its value determined?
Answer:
Yes, k-means clustering algorithm will always converge. Though, it will always converge, it is not assured that convergence will be global optimum.
Since, k-means clustering algorithm refines the value of centroid each time and check if it matches with previous value, time it will take for convergence will vary with complexity of data. Hence, there should be bound on iteration when data is complex. There is no such formula or algorithm to determine the value of maximum iterations. Though, we can calculate within-cluster squared distance error and choose the tolerence value as per our convinience.

2.2 What is the run-time of this algorithm?
Answer:
Run time complexity for k-means algorithm is $\mathcal{O}$**(n * K * i * d)**.
Where **n:** number of data points, **K:** number of clusters, **i:** number of iterations, **d:** number of iterations.

Problem 3 [50 points]

Implement Lloyd's algorithm for k-means (see algorithm k-means below) in R and call this program C_k. As you present your code explain your protocol for

3.1 initializing centroids:
As per the Lloyd's K-means Algorithm to initialize k centroids randomly, I have used sample function of R.

```
#This function will randomly initialize k number of Centroids
initialize_Centroid = function(A,k){
  centroid = A[sample(nrow(A), k),]
  return(centroid)
}
```

3.2 maintaining k centroids:
After initializing k centroids randomly, I have calculated **Euclidean Distance** between Centroid C_k and data point D_i for each data point and stored it into a vector. Next, I have assigned each data point to a cluster based on minimum Euclidean distance by assigning label. Then, calculated the mean value for every cluster based on points in respective cluster. I have followed above process in iteration until I get stable centroids or till maximum iterations.

```
#Assign point to closest centroid by updating label
updateCentroid = function(A,label,centroid){
  #Assign labels based on minimum euclidean disatance
  for (i in 1:nrow(A)) {
    cluster_dist = rep(0,nrow(centroid))
    for(j in 1:nrow(centroid)){
      cluster_dist[j] = calculate_Euclidean_Dist(A[i,],centroid[j,])
```

```
    }
    label[i] = assignLable(cluster_dist)
  }
  return(label)
}

#Calculate Euclidean Distance function
calculate_Euclidean_Dist = function(A,centroid){
  sum = 0
  for(i in 1:(length(A))){
    sum = (A[i]-centroid[i])^2 + sum
  }
  return (sqrt(sum))
}

#This function finds minimum distance between a point and cluster and return the label
assignLable = function(cluster_dist){
  return(which.min(cluster_dist))
}
```

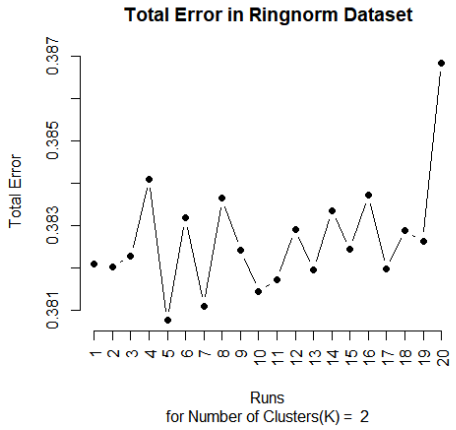
3.3 deciding ties:
If a case occured, when minimum Eucledian distance between a point and more than one centroids is same, then to decide tie, I have assigned data point D_i to the first cluster C_i with minimum value.
which.min ($distance\_from\_each\_cluster$) will return the index of first minimum value, which represents the cluster number in my implementation.

3.4 stopping criteria:
In this implementation of k-means algorithm, program will stop when centroids will become stable or maximum number of iterations crossed.
while ((!$identical(new\_centroid, old\_centroid)$) **&** ($iter \leq maximum\_iteration$))

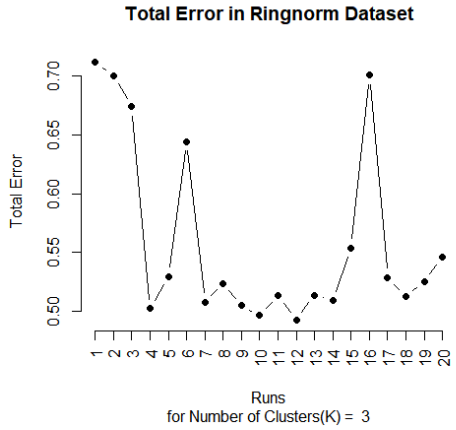
Problem 4 [50 points]

In this question, you are asked to run your program, C_k, against the Ringnorm and Ionosphere data sets and answer the following question. Report the total error rates for $k = 2, \ldots 5$ for 20 runs each, presenting the results that are easily understandable. Plots are generally a good way to convey complex ideas quickly. Discuss your results.
Answer:
Quality of a cluster is based on value of "Total Error Rate". Total Error Rate should be minimum to achieve good clustering. In this example, we have initialized centroid with random values for 20 runs. Below graphs helps us to determine, how centroid initialization is key step for generating good quality clusters. Let's study below examples:

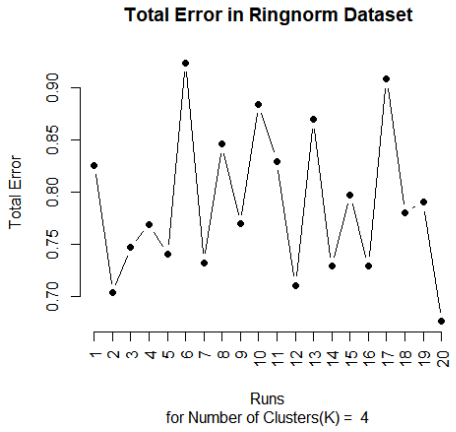
1 Ringnorm Data Set:
Figure 3 shows four graph plots for Total Error Rate when cluster size varies from 2 to 5 & over 20 runs.
As we can see in graph plot below, Total error for k=2 over 20 iteration varies from 0.381 to 0.387 which is minimum error rate among all the other values of k, which says that Quality of cluster when k=2 is best among other values. Similarly, total error rate for k=3 varies from 0.50 to 0.70 and when k=4, it varies from slightly less than 0.70 and slightly over 0.90. But, total error rate for k=5 varies from 0.85 to 1.2 which is maximum among other values of k. And hence, Quality of cluster when k=5 is worst among other values of k.



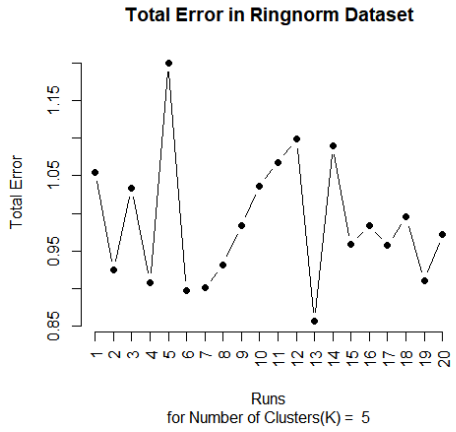
(a) Plot of Total error rates for k=2 on Ringnorm Dataset



(b) Plot of Total error rates for k=3 on Ringnorm Dataset



(c) Plot of Total error rates for k=4 on Ringnorm Dataset



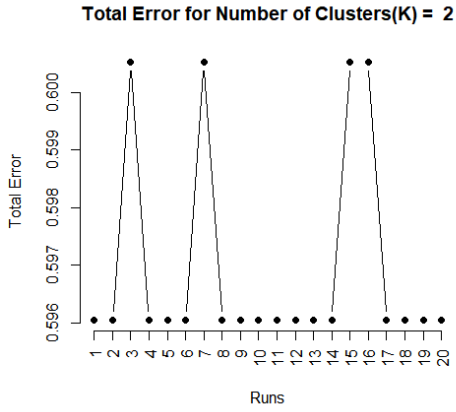
(d) Plot of Total error rates for k=5 on Ringnorm Dataset

Figure 3: Total Error in Ringnorm Dataset

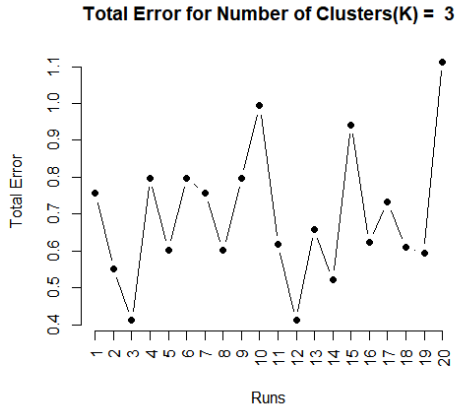
2 Ionosphere Data Set:

Figure 4 shows four graph plots for Total Error Rate when cluster size varies from 2 to 5 & over 20 runs.

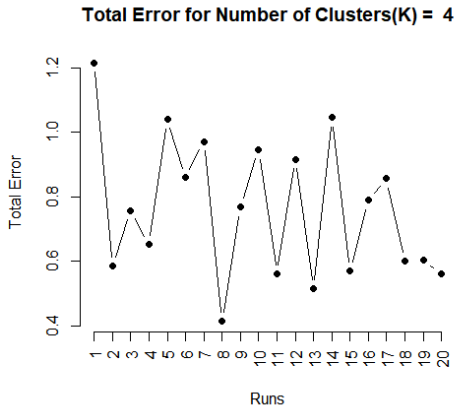
As we can see in graph plot below, Total error for k=2 over 20 iteration varies from 0.596 to slightly over 0.6 which is minimum error rate among all the other values of k, hence we can say that Quality of cluster when k=2 is best among other values of k. Total error rate for k=3 varies from 0.4 to 1.1 and when k=4, it varies from 0.4 and slightly over 1.2. But, total error rate for k=5 varies from slightly less than 0.6 to 1.2 which is maximum among other values of k. And hence, Quality of cluster when k=5 is worst among other values of k.



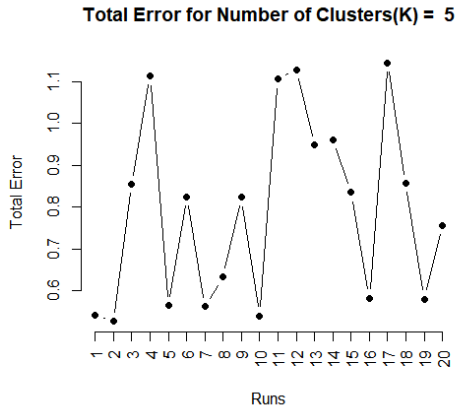
(a) Plot of Total error rates for k=2 on Ionosphere Dataset



(b) Plot of Total error rates for k=3 on Ionosphere Dataset



(c) Plot of Total error rates for k=4 on Ionosphere Dataset



(d) Plot of Total error rates for k=5 on Ionosphere Dataset

Figure 4: Total Error in Ionosphere Dataset

Problem 5 [50 points]

In this question, you are asked to make use of the R package for k-means implementation. Elbow method is one of the techniques to decide the optimal cluster number. Find the optimal number of clusters using elbow method for Ringnorm and Ionosphere data sets. Report your results in a plot as shown here for $2 \leq k \leq 10$. (The link includes an example)

Answer:

The Elbow Method is a way to find optimal number of clusters for a dataset, by plotting a graph of Total within-cluster sum of squares distance against number of clusters. When the marginal loss in total within-cluster sum of squares distance becomes minimum, we can choose that particular value of K as a "Number of Clusters". Basically, here we are trying to find optimal value of k such that, additional cluster will not give better modelling of data.

1 Ringnorm Data Set:

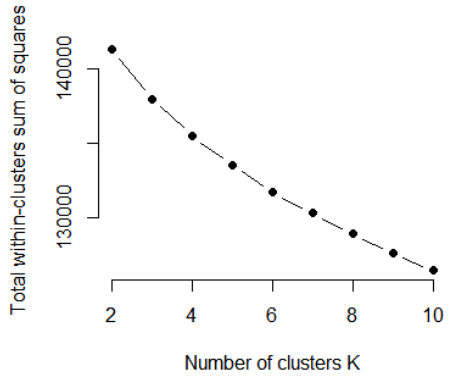


Figure 5: Elbow Method - Ringnorm Dataset

As we can see in graph plot above for Ringnorm Dataset, the total within-cluster sum of squares changes slowly till k=4, and remain very less changing thereafter. Hence, for Ringnorm Dataset, k=4 seems to be optimal choice for number of clusters. However, k=5 can also be taken as potential value for number of clusters based on Elbow method.

2 Ionosphere Data Set:

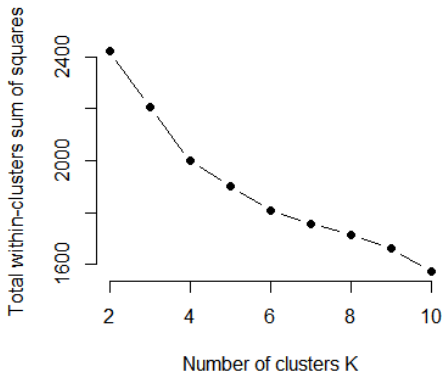


Figure 6: Elbow Method - Ionosphere Dataset

As we can see in graph plot above for Ionosphere Dataset, the total within-cluster sum of squares changes drastically till k=4, and changes slowly thereafter. Hence, forIonosphere Dataset, k=4 seems to be optimal choice for number of clusters.

Problem 6 [20 points]

Let $X \subset \mathbb{R}^n$ ($\mathbb{R}$ is the set of reals) for positive integer $n > 0$. Define a distance $d : X \times X \to \mathbb{R}_{\geq 0}$ as

$$d(x, y) = max\{|x_i - y_i|\}, \forall i \ 1 \geq i \geq n$$

Is d a metric?

Answer: According to definintion of metric space,

$$d(x, y) \geq 0$$
$$d(x, y) = 0 \leftrightarrow x = y \ldots (ReflexiveProperty)$$
$$d(x, y) = d(y, x) \ldots (SymmetricProperty)$$
$$d(x, z) \leq d(x, y) + d(y, z) \ldots (TransitiveProperty)$$

Above four conditions need to be satisfied. Let's check if $d(x, y) = max\{|x_i - y_i|\}, \forall i \ 1 \geq i \geq n$ satisfies all the above conditions.

1. d(x,y) ≥ 0
   Here, $X \subset \mathbb{R}^n$ where, $\mathbb{R}$ is the set of reals.
   $\therefore |x - y| \geq 0$
   $\therefore max\{|x_i - y_i|\} \geq 0$
   $\therefore$ d(x,y) ≥ 0 is proved.

2. d(x,y) $= 0 \leftrightarrow x = y$
   Let's assume $x_i = y_i, \forall i \ 1 \geq i \geq n$
   $\therefore |x_i - y_i| = 0, \forall i \ 1 \geq i \geq n$
   $\therefore max\{|x_i - y_i|\} = 0$
   $\therefore d(x, y) = 0 \leftrightarrow x = y$ is proved.

3. d(x,y) = d(y,x)
   $d(x, y) = max\{|x_i - y_i|\}, \forall i \ 1 \geq i \geq n$
   $\therefore d(x, y) = max\{|y_i - x_i|\}(\because | \ x - y \ |$ is an absolute value)
   $\therefore$ d(x,y) = d(y,x) is proved.

4. d(x,z) $\leq$ d(x,y) + d(y,z)
   $\therefore d(x, z) \leq max\{|x_i - y_i|\} + max\{|y_i - z_i|\}, \forall i \ 1 \geq i \geq n$
   $\therefore d(x, z) \leq max\{|x_i - y_i| + |y_i - z_i|\}$
   $\therefore d(x, z) \leq max\{|x_i - y_i + y_i - z_i|\}$
   $\therefore d(x, z) \leq max\{|x_i - z_i|\}$
   But d(x,z) $= max\{|x_i - z_i|\}$
   Hence proved.

Since, $d(x, y) = max\{|x_i - y_i|\}$ satisfies all four conditions of Equivalence Relation. It is a Distance Metric.

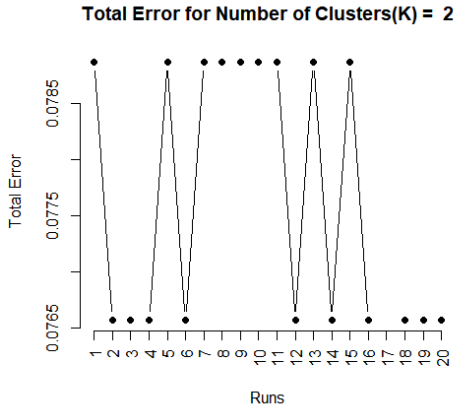
Extra credit [90 points]

This part is optional.

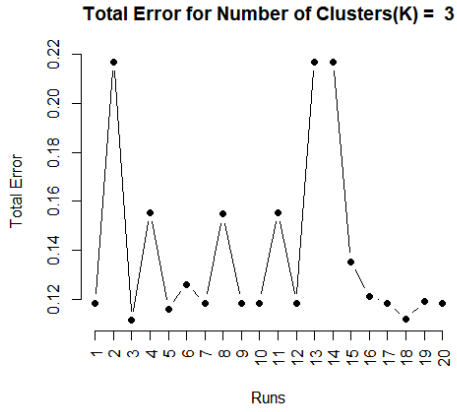
1 Answer problem 4 using Breast Cancer Wisconsin Data Set. The data sets given in Problem 4 are clean. There are no missing values on those data sets. However, Breast Cancer Wisconsin Data Set has some missing values that must be removed to use with k-means algorithm. The data set can be found here [30 points] **Answer:**

(a) Data Pre-processing:
In Breast Cancer Wisconsin Data Set, there are total 16 rows with missing values. As a part of data cleaning, I have removed those rows from the data set.
Additionally, there are total 46 Sample Code Numbers with multiple data entries (exactly 2), that is 46 women took the examination twice. Since, the data present in those entries is different, I have considered it as a valid data.

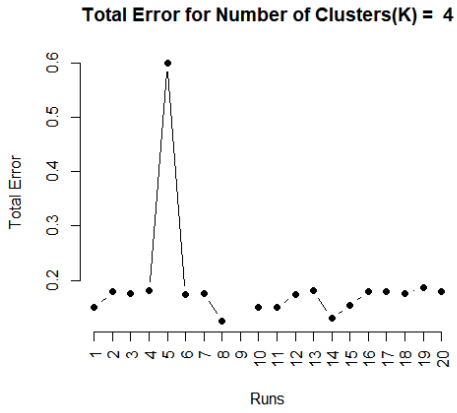
(b) Total Error Report for k=2,3..5 for 20 runs:
Figure 4 shows four graph plots for Total Error Rate when cluster size varies from 2 to 5 & over 20 runs.
As we can see in graph plot below, Total error for k=2 over 20 iteration varies from 0.0765 to slightly over 0.0785 which is minimum error rate among all the other values of k, hence we can say that Quality of cluster when k=2 is best among other values of k. Total error rate for k=3 varies from slightly below 0.12 to 0.22 and when k=4, it varies in the range of 0.2 and 0.3. But, total error rate for k=5 varies from slightly less than 0.2 to over 0.5 which is maximum among other values of k. And hence, Quality of cluster when k=5 is worst among other values of k.



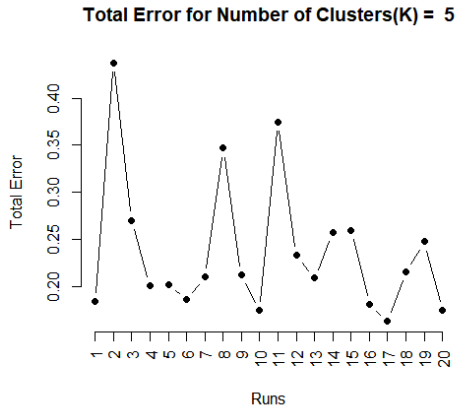
(a) Plot of Total error rates for k=2 on Breast Cancer Dataset



(b) Plot of Total error rates for k=3 on Breast Cancer Dataset



(c) Plot of Total error rates for k=4 on Breast Cancer Dataset



(d) Plot of Total error rates for k=5 on Breast Cancer Dataset

Figure 7: Total Error in Breast Cancer Dataset

2 The k-means algorithm provided above stops when centroids become stable (Line 34). In theory, k-means converges once SSE is minimized

$$SSE = \sum_{j}^{k} \sum_{x \in c_j.B} ||\mathbf{x} - c_j.v||_2^2$$

In this question, you are asked to use SSE as stopping criterion. Run k-means over Breast Cancer Wisconsin Data Set and report the total SSE in a plot for $k = 2, \ldots, 5$ for 20 runs each [30 points]. **Answer :**

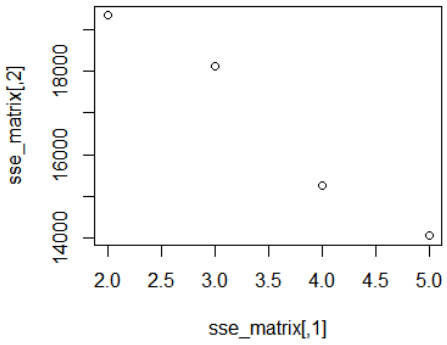


Figure 8: Plot of SSE for k=2,3..5 in Breast Cancer Wisconsin Dataset

3 Traditional k-means initialization is based on choosing values from a uniform distribution. In this question, you are asked to improve k-means through initialization. k-means ++ is an extended k-means clustering algorithm and induces non-uniform distributions over the data that serve as the initial centroids. Read the paper and discuss the idea in a paragraph. Implement this idea to improve your k-means program. [30 points]

Answer:

K-means ++ is the extention of traditional k-means clustering algorithm which focuses more on centroid initiallization. In traditional k-means clustering algorithm, we either randomly select our centroids from the dataset with uniform distribution or randomly initialize centroids to some values. In either case, the centroid initialization process can not assure the optimal solution each time. As centroid initialization is most important task in k-means clustering algorithm, we can not rely on randomness. k-means ++ is useful for non-uniform distribution of centroid over data.

Algorithm for centroid initialization:

1 Select first centroid randomly from the dataset.

2 Select closest centroid $c_i from the data point d_j and find d(c_i, d_j). Here 1 \leq i \leq centroids_i nitialized_s o_f ar$

3 Find the weighted probability distribution for each point.

4 Select the point with maximum weighted probability distribution as next centroid.

5 Repeat 2,3,4 till we initialize all the centroids.

After centroid initialization step, k-means ++ uses traditional algorithm to populate clusters with data points and to refine centroids.