Implementation Assignment 3: Kmeans clustering

Amir Azarbakht, Mandana Hamidi
School of Electrical Engineering & Computer Science
Oregon State University
1148 Kelley Engineering Center
Corvallis, OR 97331
{azarbaam,hamidim}@eecs.oregonstate.edu

Abstract

'The main objective of this assignment is to implement $Kmeans$ clustering algorithm. We did three different experiments for implementing clustering algorithm: first, we implemented the $Kmeans$ with randomly initialized cluster centers. Second, we implemented an algorithm for finding the best number of clusters. In the last part of the assignment we analyze the effect of the feature normalization on the performance of the the $Kmeans$ algorithm.

1 K means Initialization

In this section, we implemented a $Kmeans$ algorithm with $k = 5$ on the provided $data1$ with randomly initialized cluster centers for 200 times. Each time, initially, we randomly picked five unique points in the data to serve as the initial centers and ran $kmeans$ algorithm to convergence. We repeated this for 200 times and recorded the centers that were found.
$Figure$ 1 represents the recorded cluster centers over 200 runs, the original data points are plotted in blue color and the found cluster centers in the 200 runs in cyan color.

$Table$ 1 shows the values for minimum, maximum, average, and standard deviation of within-cluster sum or squared distances for the clustering found by our algorithm in the 200 random runs.

Table 1: Minimum, Maximum, Average and Standard Deviation of within-cluster sum or squared distances for the clustering found by our algorithm in the 200 random runs

Min	Max	Mean	Standard Deviation
23.0235	54.3563	25.9688	5.2813

Our results showed that the K-means algorithm found different cluster centres in different runs, mainly due to the random initialization of the initial cluster centers. This shows that the K-means algorithm is sensitive and not robust with regard to the choice of initial seeds. The best within-cluster sum of squared distances over the 200 runs was achieved to be 23.0235, which was close to the average within-cluster sum of squared distances of 25.9688. The worst within-cluster sum of squared distances over the 200 runs was

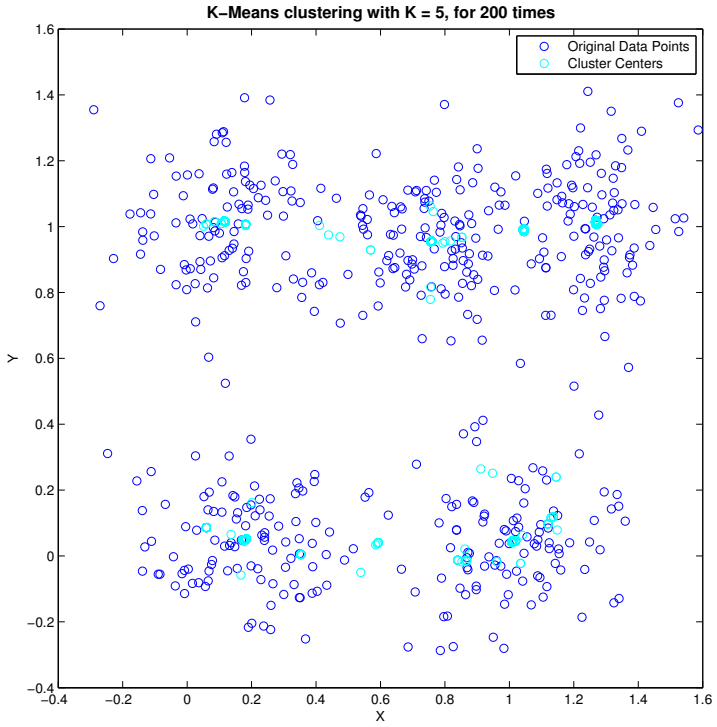

Figure 1: $Kmeans$ clustering results over 200 runs.The original data points are represented in blue color and the found cluster centers are represented in cyan color

54.3563, which was almost twice as much as the mean or minimum, which shows that the algorithm can make a really bad clustering decision because of its sensitivity to its initial seeds. The standard deviation of the within-cluster sum of squared distances over the 200 runs was 5.2813, which is almost a quarter of the mean or minimum values, which shows that the clustering decision learned by our K-means algorithm varies to some extend over 200 random runs, and this variability testifies to the sensitivity of the K-means algorithm to its random seeds.

2 Finding K

In this section, our objective was to find the best number of clusters using the K-means algorithm. We considered a variety of different k values $(k = 2, 3, ..., 15)$ and for each k value, we ran our K-means algorithm with 10 different random initializations and recorded the lowest within-cluster sum of squared distances obtained for that k value.

$Figure$ 2 shows the lowest within-cluster sum of squared distances obtained for K = 2...15. The figure shows that as K increases from 2 to 15, the lowest within-cluster sum of squared distances decreases, with a

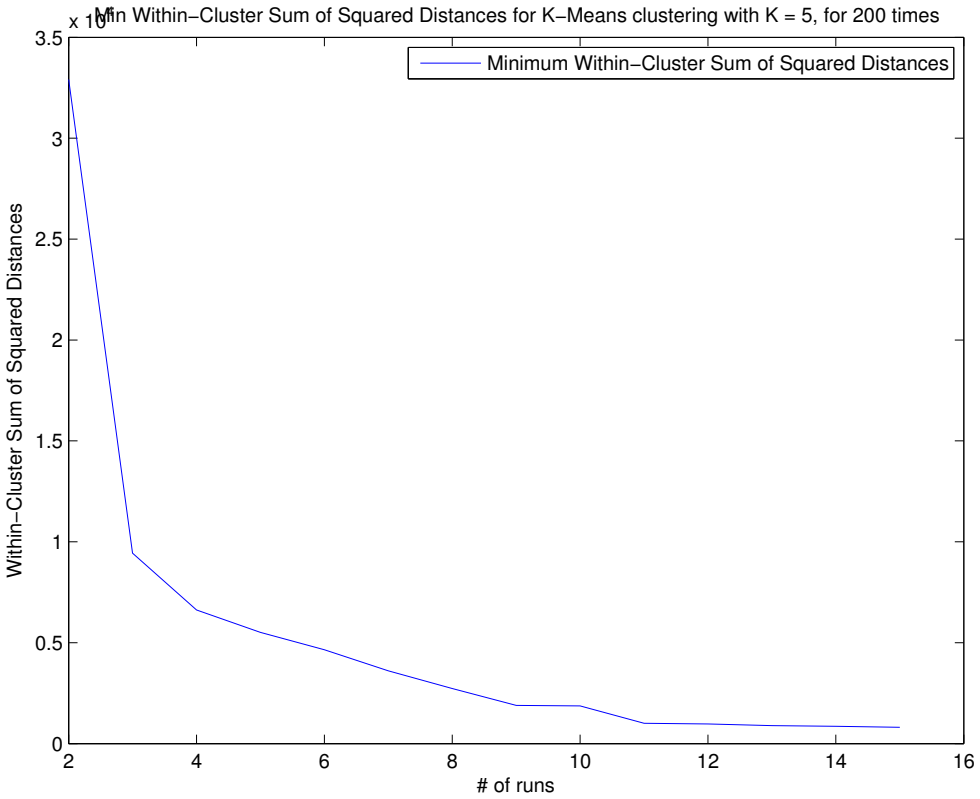

Figure 2: The lowest within-cluster sum of squared distances obtained for K = 2...15

sharp decrease in the beginning starting with K = 3 until K = 9, the first ***knee*** of the curve is located at K = 3, and the next **knee** of the curve happens to be in the interval K = [9, 10, 11], and after K = 11, the lowest within-cluster sum of squared distances remains almost steady.

As the K increases, we expect to see the lowest within-cluster sum of squared distances to decrease, because the number of clusters are increasing, and as a result we expect to see smaller clusters, and consequently a smaller lowest within-cluster sum of squared distances. This will continue to decrease to zero at the point that we have one cluster for each data point.

As for multiple knees in the data, one can argue that depending on the granularity of the clusters, and the chosen K, as $Figure$ 3 shows, we can see a knee in the graph – e.g. when K = 3. the data points are clustered into 3 clusters – which make sense to human eyes, and are reflected in the lowest sum of squared distances vs. K plot as a knee, however, if we continue to increase K, to find a more fine-grain clustering, we expect to see another knee at about K = 10, which $Figure$ 2 shows. In short, the multiple knees correspond to the multiple ways of clustering the data, e.g. in case of data2 dataset, if we choose to have three clusters or 10 clusters, both of which can be correct clustering decisions.

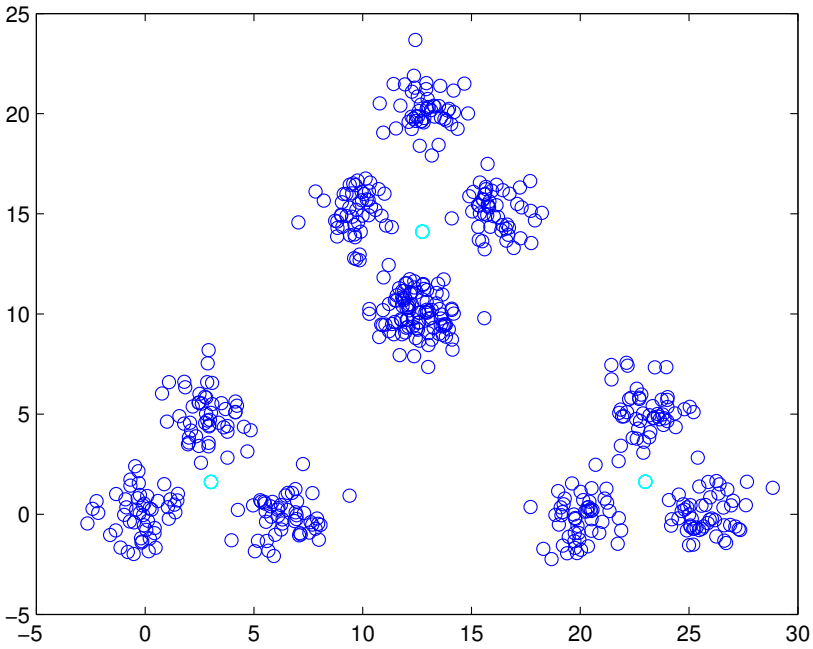

Figure 3: Data points and found cluster centers with K = 3 in data2 dataset

3 Feature normalization

In this section we analyzed the effects of feature normalization and the performance of our K-means clustering algorithm with k = 2 for 200 different random initializations. We recorded all the cluster centers that were found in these 200 runs.

$Figure\ 4$ shows the original data in blue color and the cluster centers in the same figure using a cyan color. As seen in this figure, the center of clusters are on the border of two main clusters seen in the figure.

$Figure\ 5$ shows the normalized data in blue color and the cluster centers in cyan color. As seen in this figure, the center of clusters are in the center of the two main clusters seen in the figure, which shows that our K-means clustering algorithm performed a better job given normalized data, in case of data3 dataset.

To normalize the data, we first centered the data to have zero mean(by subtracting the mean from the data), then rescaled each feature dimension to have unit variance.

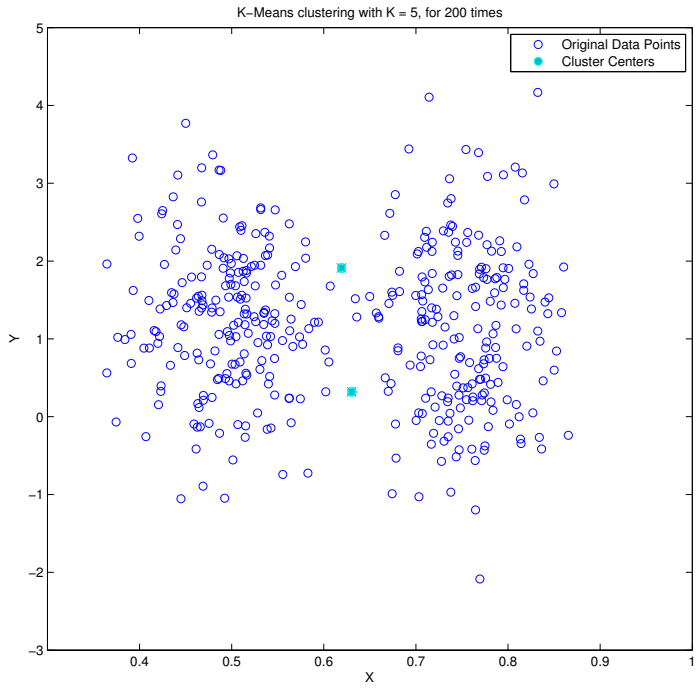

Figure 4: K-means clustering over un-normalized data for the data3 dataset with K = 2

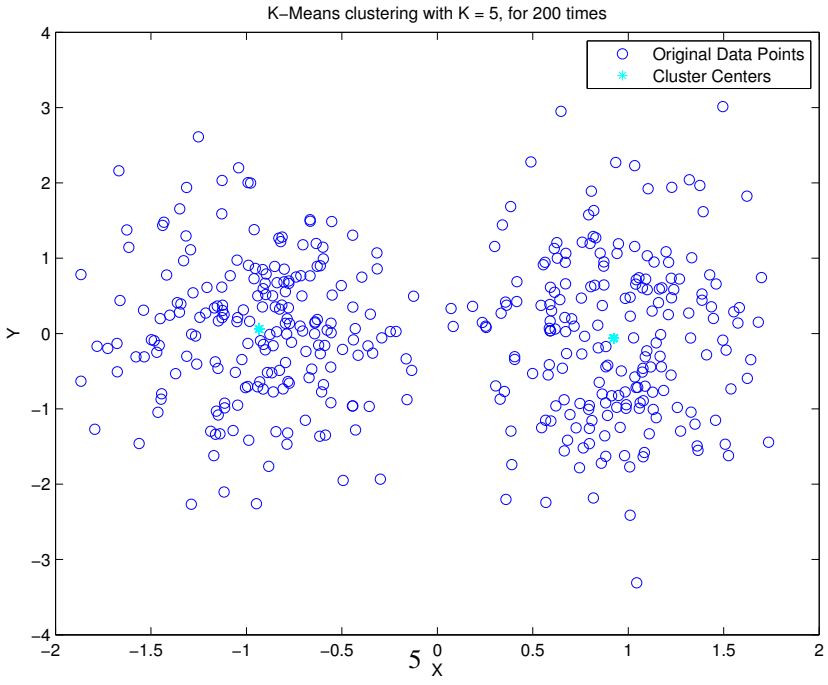

Figure 5: K-means clustering over normalized data for the data3 dataset with K = 2