Almaty Singapore

Lab 9: K-Means Clustering

Introduction:

Pattern recognition employs clustering to automatically group data based on a similarity measure. Similarly, clustering is utilized in "unsupervised learning" to identify groupings without prior knowledge of how the data should be classified; instead, a definition of how dissimilar two items are is used to identify the groups. A clustering algorithm that is both simple and efficient is called K-means. Essentially, the K-means algorithm works by identifying K centers, gathering all data closest to each center, and then moving the center to the center of the nearby data. This process is repeated with the new centers until they cease to move.

K-Means:

K-means is a clustering algorithm that aims to partition a given dataset into K clusters where each cluster represents a group of data points that are similar to each other. The algorithm starts by randomly selecting K data points from the dataset as the initial centers of the clusters. Then, for each data point in the dataset, the algorithm assigns it to the nearest cluster center based on a distance metric, such as Euclidean distance.

After all data points have been assigned to a cluster, the algorithm calculates the new centers of each cluster as the mean of all the data points belonging to that cluster. This process of assigning data points to clusters and updating the centers of the clusters is repeated until the centers of the clusters no longer change significantly or a maximum number of iterations is reached.

The K-means algorithm can be applied to various fields, including image segmentation, document clustering, and market segmentation, to name a few. It is a widely used clustering algorithm due to its simplicity and efficiency. However, it does have some limitations, such as its sensitivity to the initial random selection of the cluster centers and its tendency to converge to a local optimum instead of a global optimum.

In K-means clustering, the objective is to partition a dataset into K clusters such that the sum of squared distances between the data points and their assigned cluster centroids is minimized. This optimization problem can lead to two types of optima: local optima and global optima.

A local optimum is a clustering solution where the K-means algorithm converges to a suboptimal solution. This means that the algorithm has found a

solution that reduces the sum of squared distances between the data points and their assigned cluster centroids, but it is not the best solution overall. A local optimum can occur when the algorithm gets stuck in a solution that reduces the objective function without finding a better solution. This can happen if the initial placement of the centroids is not appropriate, or if the algorithm gets trapped in a local minimum.

A global optimum is a clustering solution where the K-means algorithm converges to the optimal solution, which is the solution that has the minimum sum of squared distances between the data points and their assigned cluster centroids. The global optimum is the ideal solution that provides the best possible clustering of the data. Achieving the global optimum is desirable but can be challenging, and various techniques can be used to improve the chances of finding it, such as using different initialization methods, running the algorithm multiple times, or using a more advanced clustering algorithm.

K-means has two challenging aspects. Firstly, determining the appropriate value for the k parameter can be difficult since increasing the number of clusters tends to result in clusters that appear more tightly packed, i.e., they have lower variance. Secondly, defining a distance metric based on the dataset's features can be challenging if the data has multiple features and/or non-numeric features. For example, if the data has a feature with the values {"A", "S", "D", "F"}, determining the distance between each of these values can be problematic.

Additionally, if the dataset has numeric features with different means and variances, deciding whether to rescale the features can be tricky. Even after addressing these issues, choosing the appropriate distance metric remains a challenge. Therefore, this study implements three highly-related distance metrics to tackle these challenges: Norm 1 (Manhattan Distance), Norm 2 (Euclidean Distance), and the Infinity Norm (Infinite Distance). Manhattan distance means that the distance between two items is the sum of the absolute values of the differences. In other words, it is the distance that one would travel along a grid-like structure to get from one point to another, by moving only horizontally and vertically along a set of axes. Euclidean distance means that the distance between two items is the square root of the sum of the squares of the distances. In other words, Euclidean distance is the straight-line distance between two points in Euclidean space. Infinite distance means that the distance between two items is the largest absolute value difference. In other words, infinite distance is a distance metric used to indicate that two points are not similar at all. The infinite distance metric is useful when dealing with data that has a binary or categorical nature. It is used to avoid situations where the similarity between two points is overestimated due to their proximity in a lower-dimensional space, which can happen when using other distance metrics like Euclidean distance or Manhattan distance.

In K-means clustering, infinite distance can be used to define a stopping criterion for the algorithm or to assign points to a separate cluster when they are too dissimilar to belong to any existing cluster.

Questions:

To determine the best value of k for the K-means clustering algorithm, we can use the elbow method. The elbow method involves plotting the number of clusters (k) against the sum of squared distances of the data points to their assigned cluster centroids. The idea is to choose the value of k at the "elbow" or bend in the plot, where adding more clusters does not significantly improve the quality of the clustering. Here is a plot of the sum of squared distances for different values of k, projected onto a 2-D plane, for the cluster1.csv dataset (Figure 1) and a pictorial representation of the clustering of the data in cluster1.csv using the data points in only the first two columns as coordinates (Figure 2):

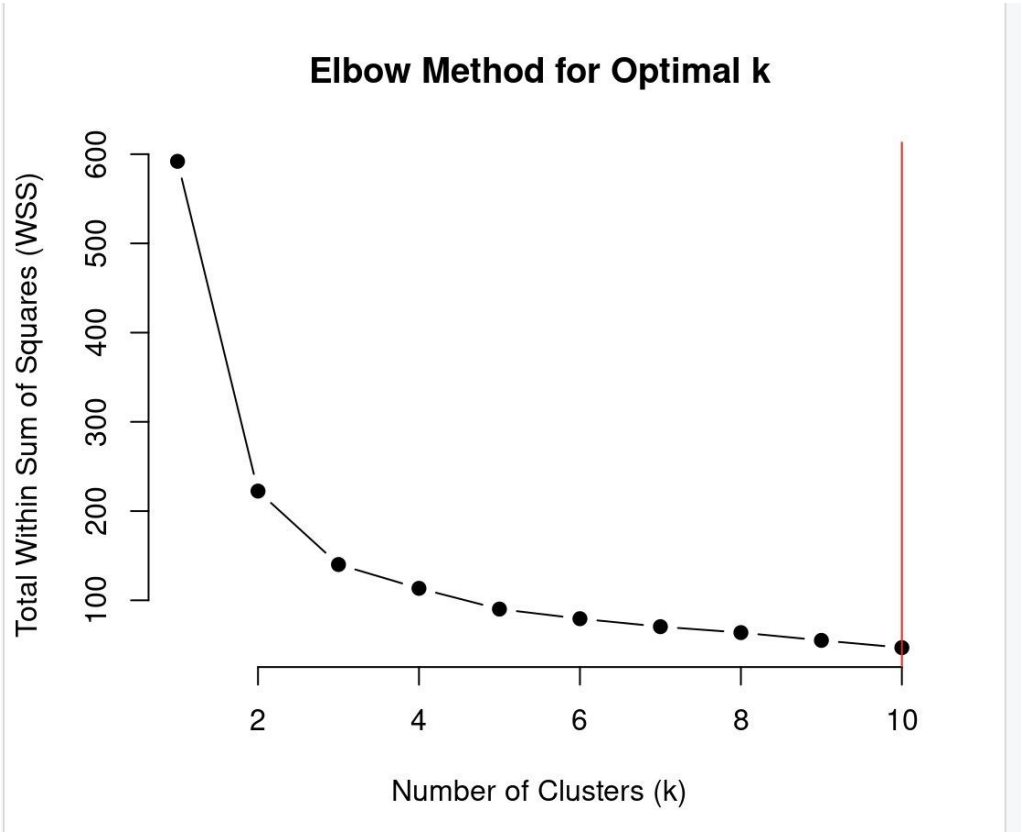

Figure 1.

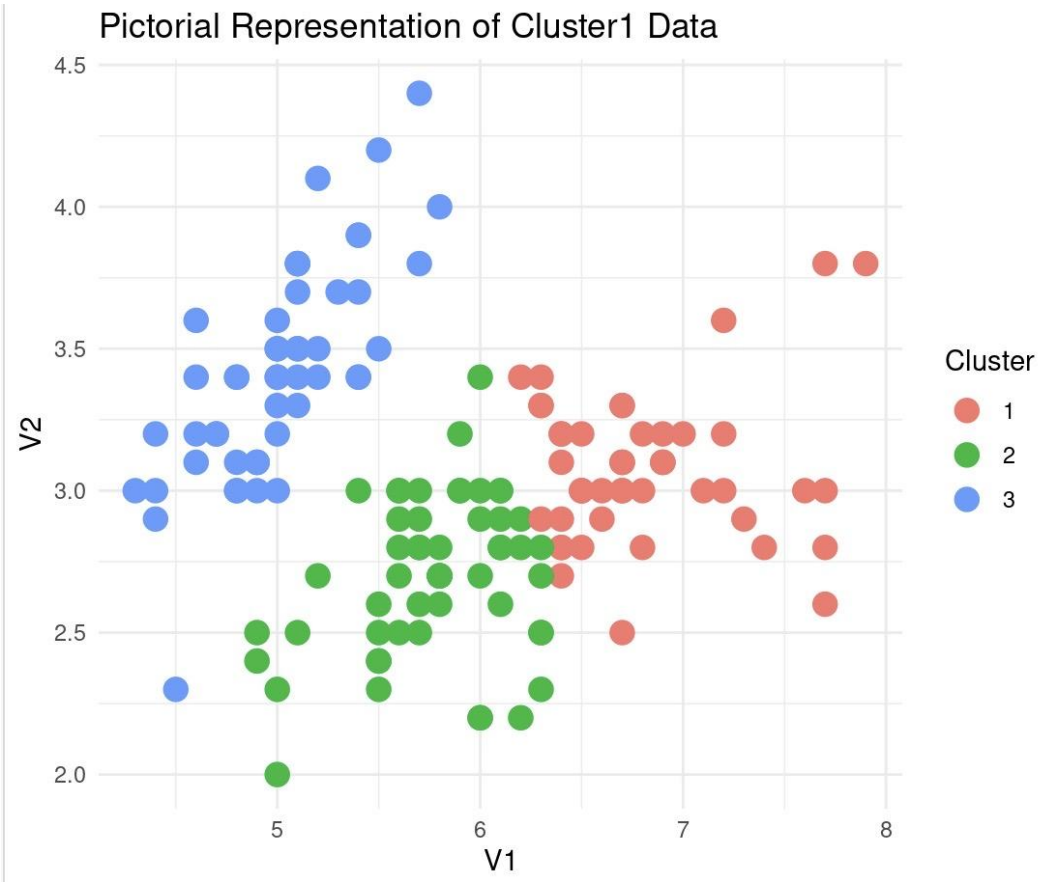

Figure 2.

Based on the elbow method, the plot suggests that the optimal number of clusters for this dataset is around 3 or 4. Beyond k=3 or k=4, the decrease in sum of squared distances becomes less significant, indicating that adding more clusters does not result in much improvement in clustering quality. The elbow point in the plot is not very sharp, which suggests that there is not a clear-cut answer for the best number of clusters, but we can consider the point where the marginal gain in the sum of squared distances starts to decrease more slowly. Therefore, I would choose k=3 as the best value for this dataset.

K-means is a randomized algorithm that initializes the centroids randomly before updating them to minimize the objective function. The initial randomization means that K-means can converge to different local optima depending on the starting centroids. The algorithm may converge to a different set of clusters even when the same dataset is used with the same number of clusters.

To mitigate this, K-means is often run multiple times with different random initializations, and the results are compared to choose the best set of clusters.

By taking the average of all the number of iterations for every trial run on the K-means clustering algorithm (listed in the "OUTPUT" section at the end of this study, the average number of iterations for the means to stabilize is 9-10 iterations.

The average number of iterations required for k-means clustering to converge and stabilize the means can depend on various factors: (1) The number of clusters (k): Generally, larger values of k require more iterations to converge as the algorithm has to optimize more centroids. (2) The initial placement of centroids: Randomly placing centroids can affect the number of iterations needed to converge. Poor initial placements may require more iterations to stabilize. (3) The size of the dataset: Larger datasets with more points may require more iterations to converge as there are more calculations to be made. (4) The distribution of the data: Highly skewed or dense distributions can cause the algorithm to converge slowly, resulting in more iterations. (5) The distance metric used: Different distance metrics can result in different convergence rates. Euclidean distance, for example, is known to converge faster than other distance metrics such as Manhattan distance. (6) The stopping criterion: The stopping criterion, which determines when the algorithm should stop iterating, can affect the number of iterations needed to converge. Tighter stopping criteria may require more iterations to reach convergence.

K-means does not always converge to the same clusters. The final clustering result can be sensitive to the initial randomly chosen centroids, which can result in the algorithm converging to different local optima. Even when the initial centroids are the same, the order in which the data points are processed can affect the convergence. However, it is important to note that the chance of convergence to different local optima decreases as the number of iterations increases. So, running the K-means algorithm for a sufficient number of iterations can increase the likelihood of convergence to the same clusters. Additionally, using a smarter initialization technique, such as randomization, can also improve the stability of the results.

Rescaling the data so that all dimensions have the same range helps with clustering systems like k-means because it puts all the features on the same scale and prevents features with larger ranges from dominating the distance calculations. In K-means clustering, the distance between observations is calculated using the Manhattan distance, Euclidean distance, or Infinite Distance formulas. When the features have different ranges, features with larger ranges will dominate the distance calculation, and the clustering algorithm will be biased towards those features. As a result, the clustering algorithm may not perform well and may lead to incorrect or suboptimal clusters. By rescaling the data, all the

features are put on the same scale, and the distance calculations are based on a more balanced representation of all the features. This can improve the performance of the clustering algorithm and help ensure that all features are equally considered in the clustering process.

When there are different variances along different dimensions, it means that some dimensions have a larger scale of measurement compared to others. This can cause the clustering algorithm to overweight dimensions with a larger scale and underweight dimensions with a smaller scale, leading to suboptimal clustering results.

For example, consider a dataset with two dimensions: age (measured in years) and income (measured in dollars). If we apply K-means clustering directly to this dataset without rescaling the features, the clustering algorithm may overweight income compared to age, because the range of income values is much larger than that of age values. This can lead to clusters being formed mainly based on income, with age having little influence on the clustering result.

One way to alleviate this issue is to rescale the features so that they have a similar scale of measurement. This can be achieved by normalizing the features to have zero mean and unit variance, or by scaling them to a specific range (e.g., [0,1]). Rescaling the features ensures that each dimension contributes equally to the clustering result, regardless of its scale of measurement.

Conclusion:

In conclusion, K-means clustering is a widely used and effective unsupervised learning algorithm for grouping data points based on their similarities. It has been shown to be particularly useful in various applications, such as image processing, market segmentation, and customer segmentation. However, the effectiveness of k-means clustering depends on several factors, including the choice of the initial centroids, the selection of the appropriate number of clusters, and the quality of the input data. Despite its limitations, k-means clustering remains a powerful tool for exploratory data analysis, and its practical applications continue to expand in various fields.

Appendix:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Random;
public class KMeansClustering {
    public static float[][] cluster(float[][] data,
int K, float closeEnough) {
        int M = data.length;
        int N = data[0].length;
        float[][] CM = new float[K][N];
        float[][] OLD_CM = new float[K][N];

        // Initialize OLD_CM to all 0
        for (int i = 0; i < K; i++) {
            for (int j = 0; j < N; j++) {
                OLD_CM[i][j] = 0;
            }
        }

        /**
        // Initialize CM to first two items in data
        for (int i = 0; i < K; i++) {
            for (int j = 0; j < N; j++) {
                CM[i][j] = data[i][j];
            }
        }
        */
        // Initialize CM to K randomly chosen elements
from DATA
        Random rand = new Random();
        for (int i = 0; i < K; i++) {
            int index = rand.nextInt(M);
          for (int j = 0; j < N; j++) {
                CM[i][j] = data[index][j];
            }
```

```
        }

          int numIterations = 0;
        while (true) {
          numIterations++;
            if (clusDistance(CM, OLD_CM) <
closeEnough) {
                break;
            }

            // Let: groups be an array of size K of
something that can hold 1xM arrays, initially empty
            float[][][] groups = new float[K][][];

            for (int i = 0; i < K; i++) {
                groups[i] = new float[0][N];
            }

            for (int i = 0; i < M; i++) {
                // Find the citem in CM to which item
is closest
                int closest = -1;
                float minDistance = Float.MAX_VALUE;

                for (int j = 0; j < K; j++) {
                    float distance =
ManhattanDistance(data[i], CM[j]);
                    if (distance < minDistance) {
                        closest = j;
                        minDistance = distance;
                    }
                }

                // Add item to group corresponding to
closest citem
```

```
                float[][] newGroup = new
float[groups[closest].length + 1][N];
                for (int j = 0; j <
groups[closest].length; j++) {
                    newGroup[j] = groups[closest][j];
                }
                newGroup[newGroup.length - 1] =
data[i];
                groups[closest] = newGroup;
            }

            // Copy CM into OLD_CM
            for (int i = 0; i < K; i++) {
                for (int j = 0; j < N; j++) {
                    OLD_CM[i][j] = CM[i][j];
                }
            }

            for (int i = 0; i < K; i++) {
                // CM[j] = centroid of group[j] (the
average of the group)
                for (int j = 0; j < N; j++) {
                    float sum = 0;
                    for (int k = 0; k <
groups[i].length; k++) {
                        sum += groups[i][k][j];
                    }
                    CM[i][j] = sum / groups[i].length;
                }
            }
        }
          System.out.println("Number of iterations" +
numIterations);
        return CM;
    }
```

```
    public static float clusDistance(float[][] cOne,
float[][] cTwo) {
       // Clus_Distance(cOne, cTwo) // cOne and cTwo
are KxM 2-D arrays
        float d = 0;
        for (int i = 0; i < cOne.length; i++) {
            d += ManhattanDistance(cOne[i], cTwo[i]);
        }
        return d;
    }
    public static float ManhattanDistance(float[]
item1, float[] item2) {
        float distance = 0;
        for (int i = 0; i < item1.length; i++) {
            distance += Math.abs(item1[i] - item2[i]);
        }
        return distance;
    }
    public static float EuclideanDistance(float[]
item1, float[] item2) {
        float sum = 0;
        for (int i = 0; i < item1.length; i++) {
            float diff = item1[i] - item2[i];
            sum += diff * diff;
        }
        return (float) Math.sqrt(sum);
    }
    public static float InfiniteDistance(float[]
item1, float[] item2) {
        float maxDiff = 0;
        for (int i = 0; i < item1.length; i++) {
            float diff = Math.abs(item1[i] -
item2[i]);
            if (diff > maxDiff) {
                maxDiff = diff;
            }
```

```
        }
        return maxDiff;
    }
    public static float[][] readCSV(String filename)
throws IOException {
        BufferedReader br = new BufferedReader(new
FileReader(filename));
        String line = "";
        int row = 0, col = 0;
        while ((line = br.readLine()) != null) {
            String[] values = line.split(",");
            row++;
            col = values.length;
        }
        br.close();
        float[][] data = new float[row][col];
        br = new BufferedReader(new
FileReader(filename));
        row = 0;
        while ((line = br.readLine()) != null) {
            String[] values = line.split(",");
            for (int i = 0; i < col; i++) {
                data[row][i] =
Float.parseFloat(values[i]);
            }
            row++;
        }
        br.close();
        return data;
    }
    public static void main(String[] args) {
        try {
            // Read the data from the CSV file
            float[][] data =
KMeansClustering.readCSV("cluster1.csv");
```

```
            // Set the number of clusters and the
threshold for convergence to 0.01
            int K = 2;
          float closeEnough = 0.01f;

            // Cluster the data
            float[][] centroids =
KMeansClustering.cluster(data, K, closeEnough);

            // Print the centroids
            System.out.println("cluster1.csv test
using Norm 1 on random starting points:");
            for (int i = 0; i < centroids.length; i++)
{
                System.out.print("  Cluster " + i + ":
[");
                for (int j = 0; j <
centroids[i].length; j++) {
                    System.out.print(centroids[i][j]);
                    if (j < centroids[i].length - 1) {
                        System.out.print(", ");
                    }
                }
                System.out.println("]");
            }
        } catch (IOException e) {
            System.err.println("Error reading CSV
file: " + e.getMessage());
        }
    }
}
```

OUTPUT:

```
Number of iterations: 4
cluster1.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [6.3145847, 2.8958337, 4.9739594, 1.7031251]
  Cluster 1: [5.005555, 3.3351853, 1.5981482, 0.30185184]

Number of iterations: 7
cluster1.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [6.3145847, 2.8958337, 4.9739594, 1.7031251]
  Cluster 1: [5.005555, 3.3351853, 1.5981482, 0.30185184]

Number of iterations: 5
cluster1.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [5.005555, 3.3351853, 1.5981482, 0.30185184]
  Cluster 1: [6.3145847, 2.8958337, 4.9739594, 1.7031251]

Number of iterations: 12
cluster1.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [6.8702683, 3.0864866, 5.745946, 2.0891893]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [5.9047623, 2.746032, 4.4126983, 1.4333335]

Number of iterations: 9
cluster1.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [4.747826, 2.9347825, 1.7695649, 0.33043477]
  Cluster 1: [6.3145847, 2.8958337, 4.9739594, 1.7031251]
  Cluster 2: [5.1967745, 3.6322582, 1.4709678, 0.2806451]

Number of iterations: 11
cluster1.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [4.747826, 2.9347825, 1.7695649, 0.33043477]
  Cluster 1: [6.3145847, 2.8958337, 4.9739594, 1.7031251]
  Cluster 2: [5.1967745, 3.6322582, 1.4709678, 0.2806451]
```

Number of iterations: 5
cluster1.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [6.2113643, 2.865909, 4.8045464, 1.6204547]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [6.932256, 3.1064515, 5.8548384, 2.1419356]
  Cluster 3: [5.5199995, 2.592, 3.908, 1.1960001]

Number of iterations: 6
cluster1.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [6.1860466, 2.8534884, 4.7813964, 1.595349]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [5.5124993, 2.5833333, 3.8833332, 1.1916667]
  Cluster 3: [6.9060593, 3.1060605, 5.812121, 2.1333334]

Number of iterations: 5
cluster1.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [6.8702683, 3.0864866, 5.745946, 2.0891893]
  Cluster 1: [4.7130437, 3.121739, 1.4173912, 0.19130436]
  Cluster 2: [5.9047623, 2.746032, 4.4126983, 1.4333335]
  Cluster 3: [5.255555, 3.6703706, 1.5037035, 0.28888884]

Number of iterations: 8
cluster1.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [7.54, 3.1499999, 6.3900003, 2.09]
  Cluster 1: [6.1900005, 2.8575, 4.7550006, 1.59]
  Cluster 2: [6.608, 3.064, 5.5519996, 2.1279998]
  Cluster 3: [5.5199995, 2.592, 3.908, 1.1960001]
  Cluster 4: [5.0059996, 3.4180002, 1.464, 0.24399997]

Number of iterations: 17
cluster1.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [6.2787237, 2.8957446, 4.934043, 1.708511]
  Cluster 1: [7.079166, 3.125, 5.975, 2.1499999]
  Cluster 2: [5.255555, 3.6703706, 1.5037035, 0.28888884]
  Cluster 3: [5.5586205, 2.6241379, 3.975862, 1.2310345]
  Cluster 4: [4.7130437, 3.121739, 1.4173912, 0.19130436]

Number of iterations: 11
cluster1.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [4.7130437, 3.121739, 1.4173912, 0.19130436]
  Cluster 1: [6.9060593, 3.1060605, 5.812121, 2.1333334]
  Cluster 2: [6.1860466, 2.8534884, 4.7813964, 1.595349]
  Cluster 3: [5.255555, 3.6703706, 1.5037035, 0.28888884]
  Cluster 4: [5.5124993, 2.5833333, 3.8833332, 1.1916667]

Number of iterations: 9
cluster1.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [6.568182, 3.0863638, 5.5363636, 2.1636362]
  Cluster 1: [6.063158, 2.7473683, 5.0263157, 1.7894735]
  Cluster 2: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 3: [5.4772725, 2.5590909, 3.8636363, 1.1818182]
  Cluster 4: [7.4750004, 3.125, 6.2999997, 2.05]
  Cluster 5: [6.2520003, 2.9320002, 4.5080004, 1.4159999]

Number of iterations: 9
cluster1.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [4.8181825, 3.221212, 1.4363637, 0.22727269]
  Cluster 1: [5.370588, 3.8, 1.5176469, 0.27647057]
  Cluster 2: [5.5199995, 2.592, 3.908, 1.1960001]
  Cluster 3: [6.56087, 3.069565, 5.5260863, 2.1521738]
  Cluster 4: [6.1900005, 2.8575, 4.7550006, 1.59]
  Cluster 5: [7.4750004, 3.125, 6.2999997, 2.05]

Number of iterations: 8
cluster1.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [5.370588, 3.8, 1.5176469, 0.27647057]
  Cluster 1: [5.3923078, 2.4384615, 3.6538463, 1.123077]
  Cluster 2: [6.985713, 3.1071427, 5.899999, 2.1607144]
  Cluster 3: [4.8181825, 3.221212, 1.4363637, 0.22727269]
  Cluster 4: [6.4249997, 2.964286, 4.9071436, 1.6714284]
  Cluster 5: [5.8258066, 2.7580645, 4.532258, 1.4741937]

Number of iterations: 5
cluster1.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [6.3010325, 2.8865983, 4.9587636, 1.6958765]
  Cluster 1: [5.00566, 3.3603776, 1.5622642, 0.28867924]

Number of iterations: 6
cluster1.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [5.00566, 3.3603776, 1.5622642, 0.28867924]
  Cluster 1: [6.3010325, 2.8865983, 4.9587636, 1.6958765]

Number of iterations: 5
cluster1.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [6.3010325, 2.8865983, 4.9587636, 1.6958765]
  Cluster 1: [5.00566, 3.3603776, 1.5622642, 0.28867924]

Number of iterations: 10
cluster1.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [6.853844, 3.0769234, 5.715385, 2.0538464]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [5.8836064, 2.7409837, 4.388525, 1.4344264]

Number of iterations: 7
cluster1.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [5.901613, 2.7483873, 4.393549, 1.4338712]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [6.849999, 3.0736842, 5.742105, 2.0710528]

Number of iterations: 8
cluster1.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [5.8836064, 2.7409837, 4.388525, 1.4344264]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [6.853844, 3.0769234, 5.715385, 2.0538464]

Number of iterations: 7
cluster1.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 1: [5.5321426, 2.6357143, 3.960714, 1.2285715]
  Cluster 2: [6.9499984, 3.1066666, 5.8666663, 2.1533332]
  Cluster 3: [6.2571435, 2.8619046, 4.850001, 1.6333334]
Number of iterations: 21
cluster1.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [6.853844, 3.0769234, 5.715385, 2.0538464]
  Cluster 1: [5.370588, 3.8, 1.5176469, 0.27647057]
  Cluster 2: [5.8836064, 2.7409837, 4.388525, 1.4344264]
  Cluster 3: [4.8181825, 3.221212, 1.4363637, 0.22727269]

Number of iterations: 5
cluster1.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [6.849999, 3.0736842, 5.742105, 2.0710528]
  Cluster 1: [5.901613, 2.7483873, 4.393549, 1.4338712]
  Cluster 2: [5.255555, 3.6703706, 1.5037035, 0.28888884]
  Cluster 3: [4.7130437, 3.121739, 1.4173912, 0.19130436]

Number of iterations: 5
cluster1.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [5.4125, 2.46875, 3.74375, 1.16875]
  Cluster 1: [6.288889, 2.9055557, 5.1111116, 1.852778]
  Cluster 2: [6.0192304, 2.869231, 4.384615, 1.3576921]
  Cluster 3: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 4: [7.1227264, 3.1136363, 6.031818, 2.131818]

Number of iterations: 12
cluster1.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [7.1227264, 3.1136363, 6.031818, 2.131818]
  Cluster 1: [6.423809, 2.9190478, 4.604762, 1.4380951]
  Cluster 2: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 3: [6.196552, 2.8827586, 5.1827583, 1.9344827]
  Cluster 4: [5.5321426, 2.6357143, 3.960714, 1.2285715]

Number of iterations: 8
cluster1.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [6.912498, 3.1, 5.8468747, 2.13125]
  Cluster 1: [4.7250004, 3.1333332, 1.4208335, 0.19166666]
  Cluster 2: [5.5321426, 2.6357143, 3.960714, 1.2285715]
  Cluster 3: [6.2525005, 2.855, 4.8150005, 1.6250002]
  Cluster 4: [5.265384, 3.6807694, 1.503846, 0.29230765]

Number of iterations: 14
cluster1.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [6.5291667, 3.0583332, 5.5083327, 2.1624997]
  Cluster 1: [5.5079994, 2.6, 3.908, 1.204]
  Cluster 2: [5.265384, 3.6807694, 1.503846, 0.29230765]
  Cluster 3: [4.7250004, 3.1333332, 1.4208335, 0.19166666]
  Cluster 4: [7.4750004, 3.125, 6.2999997, 2.05]
  Cluster 5: [6.2076926, 2.8538463, 4.746154, 1.5641025]

Number of iterations: 7
cluster1.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [7.0148134, 3.096296, 5.918518, 2.1555555]
  Cluster 1: [5.121739, 3.5173917, 1.5304348, 0.27826086]
  Cluster 2: [5.5285716, 4.042857, 1.4714285, 0.2857143]
  Cluster 3: [4.69, 3.0849998, 1.385, 0.19000001]
  Cluster 4: [6.2644444, 2.8844442, 4.8866673, 1.6666669]
  Cluster 5: [5.5321426, 2.6357143, 3.960714, 1.2285715]

Number of iterations: 6
cluster1.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [5.6285706, 2.7238095, 4.133333, 1.2952381]
  Cluster 1: [6.423809, 2.9190478, 4.604762, 1.4380951]
  Cluster 2: [6.196552, 2.8827586, 5.1827583, 1.9344827]
  Cluster 3: [5.2428565, 2.3714283, 3.4428573, 1.0285714]
  Cluster 4: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 5: [7.1227264, 3.1136363, 6.031818, 2.131818]

Number of iterations: 3
cluster1.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [6.273739, 2.875758, 4.9252534, 1.6818184]
  Cluster 1: [5.007843, 3.4, 1.4941176, 0.2607843]

Number of iterations: 4
cluster1.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [6.273739, 2.875758, 4.9252534, 1.6818184]
  Cluster 1: [5.007843, 3.4, 1.4941176, 0.2607843]

Number of iterations: 4
cluster1.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [6.273739, 2.875758, 4.9252534, 1.6818184]
  Cluster 1: [5.007843, 3.4, 1.4941176, 0.2607843]

Number of iterations: 6
cluster1.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [5.2759995, 3.6600003, 1.5519998, 0.31199998]
  Cluster 1: [4.7500005, 3.1499999, 1.4384618, 0.21153846]
  Cluster 2: [6.273739, 2.875758, 4.9252534, 1.6818184]

Number of iterations: 5
cluster1.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [6.3010325, 2.8865983, 4.9587636, 1.6958765]
  Cluster 1: [4.6952386, 3.0999997, 1.395238, 0.1904762]
  Cluster 2: [5.209375, 3.5312505, 1.6718749, 0.35312498]

Number of iterations: 5
cluster1.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [6.273739, 2.875758, 4.9252534, 1.6818184]
  Cluster 1: [5.2833333, 3.7083337, 1.4916664, 0.27916664]
  Cluster 2: [4.7629633, 3.1259258, 1.4962965, 0.24444444]

Number of iterations: 5
cluster1.csv test using Norm infinity on random starting points for K = 4:
  Cluster 0: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 1: [5.5766664, 2.6300004, 3.99, 1.24]
  Cluster 2: [6.318, 2.918, 4.9820004, 1.7540004]
  Cluster 3: [7.149999, 3.12, 6.0899997, 2.1349998]

Number of iterations: 8
cluster1.csv test using Norm infinity on random starting points for K = 4:
  Cluster 0: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 1: [5.555172, 2.6206899, 3.97931, 1.2379311]
  Cluster 2: [6.2785716, 2.892857, 4.852381, 1.6761907]
  Cluster 3: [6.9448256, 3.0931034, 5.9103446, 2.113793]

Number of iterations: 18
cluster1.csv test using Norm infinity on random starting points for K = 4:
  Cluster 0: [5.5766664, 2.6300004, 3.99, 1.24]
  Cluster 1: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 2: [6.318, 2.918, 4.9820004, 1.7540004]
  Cluster 3: [7.149999, 3.12, 6.0899997, 2.1349998]

Number of iterations: 17
cluster1.csv test using Norm infinity on random starting points for K = 5:
  Cluster 0: [5.1363635, 3.5227277, 1.5272727, 0.28181818]
  Cluster 1: [4.6952386, 3.0999997, 1.395238, 0.1904762]
  Cluster 2: [5.8610168, 2.7389832, 4.369492, 1.4338986]
  Cluster 3: [6.8390236, 3.0634148, 5.678049, 2.0243905]
  Cluster 4: [5.5285716, 4.042857, 1.4714285, 0.2857143]

Number of iterations: 7
cluster1.csv test using Norm infinity on random starting points for K = 5:
  Cluster 0: [5.555172, 2.6206899, 3.97931, 1.2379311]
  Cluster 1: [5.2392855, 3.6607146, 1.517857, 0.28571424]
  Cluster 2: [6.9448256, 3.0931034, 5.9103446, 2.113793]
  Cluster 3: [4.709091, 3.1090906, 1.3954545, 0.19090909]
  Cluster 4: [6.2785716, 2.892857, 4.852381, 1.6761907]

Number of iterations: 10
cluster1.csv test using Norm infinity on random starting points for K = 5:
  Cluster 0: [6.964284, 3.0892854, 5.9321427, 2.1071427]
  Cluster 1: [5.3428574, 2.4785712, 3.707143, 1.1785716]
  Cluster 2: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 3: [6.3515153, 2.9545457, 4.9303036, 1.7515152]
  Cluster 4: [5.872, 2.74, 4.396, 1.3719999]

Number of iterations: 7
cluster1.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [4.607143, 3.0428574, 1.3857143, 0.20714287]
  Cluster 1: [5.353333, 3.8533332, 1.5133332, 0.3]
  Cluster 2: [6.8527765, 3.075, 5.786111, 2.0972223]
  Cluster 3: [5.747369, 2.6447368, 4.4736843, 1.4842105]
  Cluster 4: [6.196154, 2.923077, 4.3192306, 1.3730769]
  Cluster 5: [5.023809, 3.3571432, 1.4809523, 0.22857144]

Number of iterations: 13
cluster1.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [7.4750004, 3.125, 6.2999997, 2.05]
  Cluster 1: [5.2392855, 3.6607146, 1.517857, 0.28571424]
  Cluster 2: [6.1424246, 2.8363638, 4.724243, 1.5787878]
  Cluster 3: [4.709091, 3.1090906, 1.3954545, 0.19090909]
  Cluster 4: [6.55862, 3.037931, 5.4206886, 2.0448275]
  Cluster 5: [5.5230765, 2.6153846, 3.9192305, 1.2153846]

Number of iterations: 7
cluster1.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [6.56087, 3.069565, 5.5260863, 2.1521738]
  Cluster 1: [6.4941173, 2.8235295, 4.629412, 1.4235294]
  Cluster 2: [5.0059996, 3.4180002, 1.464, 0.24399997]
  Cluster 3: [5.9818187, 2.8681817, 4.8772726, 1.7136364]
  Cluster 4: [7.4750004, 3.125, 6.2999997, 2.05]
  Cluster 5: [5.5230765, 2.6153846, 3.9192305, 1.2153846]

Number of iterations: 11
cluster2.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [18.097343, 16.02595, 0.8839583, 6.113063, 3.6558602, 3.477723, 5.951849]
  Cluster 1: [12.887712, 13.674808, 0.86318314, 5.3363347, 3.0190382, 3.8343668, 5.080145]

Number of iterations: 7
cluster2.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [12.989191, 13.719778, 0.86402345, 5.3496976, 3.033956, 3.8160899, 5.09033]
  Cluster 1: [18.26284, 16.102161, 0.8838176, 6.140986, 3.671472, 3.487216, 5.9920278]

Number of iterations: 8
cluster2.csv test using Norm 1 on random starting points for K = 2:
  Cluster 0: [12.887712, 13.674808, 0.86318314, 5.3363347, 3.0190382, 3.8343668, 5.080145]
  Cluster 1: [18.097343, 16.02595, 0.8839583, 6.113063, 3.6558602, 3.477723, 5.951849]

Number of iterations: 13
cluster2.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [18.74917, 16.312662, 0.88470674, 6.2152157, 3.7233326, 3.6143837, 6.0758]
  Cluster 1: [14.904177, 14.574179, 0.8811612, 5.6033735, 3.3118362, 2.7978237, 5.233896]
  Cluster 2: [11.981326, 13.2797575, 0.8528855, 5.2247343, 2.879687, 4.4906626, 5.065976]

Number of iterations: 11
cluster2.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [14.904177, 14.574179, 0.8811612, 5.6033735, 3.3118362, 2.7978237, 5.233896]
  Cluster 1: [11.981326, 13.2797575, 0.8528855, 5.2247343, 2.879687, 4.4906626, 5.065976]
  Cluster 2: [18.74917, 16.312662, 0.88470674, 6.2152157, 3.7233326, 3.6143837, 6.0758]

Number of iterations: 9
cluster2.csv test using Norm 1 on random starting points for K = 3:
  Cluster 0: [14.986321, 14.614853, 0.8810058, 5.617839, 3.319486, 2.8438704, 5.2575593]
  Cluster 1: [11.9953575, 13.282975, 0.85345703, 5.2237015, 2.8835123, 4.4665, 5.062608]
  Cluster 2: [18.815517, 16.342583, 0.8846708, 6.2273784, 3.7304652, 3.5943627, 6.084862]

Number of iterations: 8
cluster2.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [18.924911, 16.38309, 0.88556725, 6.2408366, 3.7447088, 3.5431643, 6.0983453]
  Cluster 1: [13.564131, 13.943262, 0.87629765, 5.390522, 3.1457605, 3.0133936, 5.004652]
  Cluster 2: [15.683407, 14.960001, 0.8803455, 5.7361817, 3.3937278, 3.000343, 5.4401813]
  Cluster 3: [11.739843, 13.180768, 0.848594, 5.206, 2.8356771, 4.7928767, 5.0877533]

Number of iterations: 5
cluster2.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [11.889604, 13.2525, 0.84996206, 5.227447, 2.8580792, 4.6441836, 5.092592]
  Cluster 1: [16.354689, 15.283437, 0.8796, 5.846344, 3.4620311, 3.6729686, 5.6508117]
  Cluster 2: [19.12, 16.459183, 0.88668585, 6.2672653, 3.768612, 3.4729805, 6.1258774]
  Cluster 3: [14.229057, 14.239436, 0.881468, 5.4816422, 3.2386043, 2.5730793, 5.050263]

Number of iterations: 8
cluster2.csv test using Norm 1 on random starting points for K = 4:
  Cluster 0: [14.229057, 14.239436, 0.881468, 5.4816422, 3.2386043, 2.5730793, 5.050263]
  Cluster 1: [19.12, 16.459183, 0.88668585, 6.2672653, 3.768612, 3.4729805, 6.1258774]
  Cluster 2: [11.889604, 13.2525, 0.84996206, 5.227447, 2.8580792, 4.6441836, 5.092592]
  Cluster 3: [16.354689, 15.283437, 0.8796, 5.846344, 3.4620311, 3.6729686, 5.6508117]

Number of iterations: 13
cluster2.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [13.374474, 13.85079, 0.8754132, 5.357552, 3.1186056, 2.477187, 4.9733415]
  Cluster 1: [18.887678, 16.370354, 0.88515174, 6.2365, 3.7400532, 3.5633755, 6.0943923]
  Cluster 2: [15.517288, 14.882083, 0.8800354, 5.7126884, 3.3741457, 3.0379188, 5.398228]
  Cluster 3: [12.88579, 13.641581, 0.869679, 5.302158, 3.0587368, 5.801947, 5.124474]
  Cluster 4: [11.477141, 13.078572, 0.8430591, 5.1879797, 2.7812653, 4.6388364, 5.080449]

Number of iterations: 13
cluster2.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [13.564131, 13.943262, 0.87629765, 5.390522, 3.1457605, 3.0133936, 5.004652]
  Cluster 1: [18.292822, 16.098461, 0.88661015, 6.1290255, 3.684616, 3.4001026, 5.997744]
  Cluster 2: [11.739843, 13.180768, 0.848594, 5.206, 2.8356771, 4.7928767, 5.0877533]
  Cluster 3: [15.600973, 14.913171, 0.8812317, 5.7182927, 3.388195, 2.8881974, 5.411097]
  Cluster 4: [19.88842, 16.843683, 0.8806895, 6.429264, 3.8245792, 3.993106, 6.2636843]

Number of iterations: 9
cluster2.csv test using Norm 1 on random starting points for K = 5:
  Cluster 0: [19.12, 16.459183, 0.88668585, 6.2672653, 3.768612, 3.4729805, 6.1258774]
  Cluster 1: [11.836791, 13.239434, 0.8475907, 5.2323027, 2.845981, 5.328811, 5.121868]
  Cluster 2: [12.2943735, 13.371876, 0.8635719, 5.216656, 2.9485312, 2.9254408, 4.96425]
  Cluster 3: [14.476819, 14.370227, 0.88087046, 5.5356364, 3.2652273, 2.5747747, 5.0996814]
  Cluster 4: [16.354689, 15.283437, 0.8796, 5.846344, 3.4620311, 3.6729686, 5.6508117]

Number of iterations: 12
cluster2.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [19.12, 16.459183, 0.88668585, 6.2672653, 3.768612, 3.4729805, 6.1258774]
  Cluster 1: [16.354689, 15.283437, 0.8796, 5.846344, 3.4620311, 3.6729686, 5.6508117]
  Cluster 2: [12.378965, 13.404139, 0.86551726, 5.224414, 2.9653106, 2.885383, 4.9604144]
  Cluster 3: [11.438096, 13.082619, 0.8396546, 5.1984763, 2.7668095, 4.9656425, 5.1128335]
  Cluster 4: [13.044001, 13.704667, 0.87236667, 5.3198657, 3.0858662, 6.032867, 5.1150002]
  Cluster 5: [14.481396, 14.374884, 0.88055587, 5.538861, 3.2644653, 2.4791882, 5.101977]

Number of iterations: 10
cluster2.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [12.171561, 13.306875, 0.8632625, 5.19325, 2.9350626, 3.0199378, 4.9544063]
  Cluster 1: [11.7865305, 13.234693, 0.84479195, 5.2402654, 2.830857, 5.397693, 5.1379185]
  Cluster 2: [14.037186, 14.153751, 0.8807125, 5.458594, 3.211312, 3.0228155, 5.038563]
  Cluster 3: [19.246367, 16.51818, 0.8862228, 6.2954097, 3.7818635, 3.5299776, 6.144977]
  Cluster 4: [17.126818, 15.633636, 0.88027287, 5.9544086, 3.5420454, 3.7762725, 5.8440003]
  Cluster 5: [15.423549, 14.821613, 0.8821903, 5.6891937, 3.3736775, 2.606132, 5.3295155]

Number of iterations: 6
cluster2.csv test using Norm 1 on random starting points for K = 6:
  Cluster 0: [12.555925, 13.475187, 0.86854446, 5.236592, 2.9995558, 2.6149669, 4.949815]
  Cluster 1: [15.0765, 14.640999, 0.8834801, 5.61245, 3.3339496, 1.5961552, 5.16895]
  Cluster 2: [11.786949, 13.215763, 0.84724426, 5.226543, 2.8380003, 5.1379833, 5.117966]
  Cluster 3: [14.411665, 14.332999, 0.88143, 5.5106997, 3.2626333, 3.4360664, 5.092233]
  Cluster 4: [16.554232, 15.406153, 0.87610775, 5.9139614, 3.4699228, 3.9304616, 5.7629232]
  Cluster 5: [19.151043, 16.469166, 0.88708967, 6.2688546, 3.7729375, 3.4604177, 6.1272507]

Number of iterations: 6
cluster2.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [18.20329, 16.068552, 0.88462096, 6.1205916, 3.6695256, 3.4050276, 5.9584746]
  Cluster 1: [12.944255, 13.703282, 0.86327225, 5.349454, 3.0255451, 3.8676128, 5.095902]

Number of iterations: 6
cluster2.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [12.944255, 13.703282, 0.86327225, 5.349454, 3.0255451, 3.8676128, 5.095902]
  Cluster 1: [18.20329, 16.068552, 0.88462096, 6.1205916, 3.6695256, 3.4050276, 5.9584746]

Number of iterations: 6
cluster2.csv test using Norm 2 on random starting points for K = 2:
  Cluster 0: [17.942383, 15.953336, 0.8842096, 6.086416, 3.6414876, 3.4228466, 5.904192]
  Cluster 1: [12.784287, 13.629918, 0.86219114, 5.3232765, 3.003349, 3.8851042, 5.077325]

Number of iterations: 15
cluster2.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [11.964415, 13.274805, 0.8522002, 5.2292852, 2.8729222, 4.7597394, 5.088519]
  Cluster 1: [18.721804, 16.297373, 0.88508695, 6.208934, 3.7226715, 3.6035905, 6.0660987]
  Cluster 2: [14.648472, 14.460415, 0.87916666, 5.5637784, 3.2779036, 2.6489332, 5.19232]

Number of iterations: 14
cluster2.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [14.819103, 14.537164, 0.88052243, 5.5910153, 3.299359, 2.706585, 5.2175374]
  Cluster 1: [11.988659, 13.284389, 0.8527366, 5.2274265, 2.8800857, 4.583926, 5.0742435]
  Cluster 2: [18.721804, 16.297373, 0.88508695, 6.208934, 3.7226715, 3.6035905, 6.0660987]

Number of iterations: 14
cluster2.csv test using Norm 2 on random starting points for K = 3:
  Cluster 0: [14.648472, 14.460415, 0.87916666, 5.5637784, 3.2779036, 2.6489332, 5.19232]
  Cluster 1: [18.721804, 16.297373, 0.88508695, 6.208934, 3.7226715, 3.6035905, 6.0660987]
  Cluster 2: [11.964415, 13.274805, 0.8522002, 5.2292852, 2.8729222, 4.7597394, 5.088519]

Number of iterations: 6
cluster2.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [13.722067, 14.0310335, 0.87480503, 5.4274135, 3.152517, 2.5270534, 5.0425854]
  Cluster 1: [19.12, 16.459183, 0.88668585, 6.2672653, 3.768612, 3.4729805, 6.1258774]
  Cluster 2: [16.234324, 15.212703, 0.8812729, 5.8170815, 3.4594324, 3.4391105, 5.583378]
  Cluster 3: [11.887119, 13.246667, 0.8502472, 5.225364, 2.8606064, 5.0462117, 5.0980606]

Number of iterations: 12
cluster2.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [15.719087, 14.976592, 0.88038635, 5.7451825, 3.3985002, 3.1870477, 5.4483857]
  Cluster 1: [13.387446, 13.864681, 0.87411267, 5.3666162, 3.11334, 2.4528532, 4.992744]
  Cluster 2: [18.962965, 16.396666, 0.8859519, 6.242722, 3.7499256, 3.5403337, 6.1007776]
  Cluster 3: [11.894306, 13.252615, 0.8499694, 5.228708, 2.8607695, 5.0823073, 5.105615]

Number of iterations: 13
cluster2.csv test using Norm 2 on random starting points for K = 4:
  Cluster 0: [19.151043, 16.469166, 0.88708967, 6.2688546, 3.7729375, 3.4604177, 6.1272507]
  Cluster 1: [16.408066, 15.319354, 0.87826455, 5.863968, 3.4633226, 3.8500967, 5.6904507]
  Cluster 2: [14.100678, 14.195594, 0.8782237, 5.475576, 3.2125428, 2.370054, 5.0658464]
  Cluster 3: [11.91861, 13.256805, 0.85122234, 5.2256255, 2.8653197, 4.8855, 5.087472]

Number of iterations: 14
cluster2.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [19.151043, 16.469166, 0.88708967, 6.2688546, 3.7729375, 3.4604177, 6.1272507]
  Cluster 1: [14.641021, 14.448776, 0.8809735, 5.5620003, 3.280245, 2.3963711, 5.149775]
  Cluster 2: [16.517778, 15.369629, 0.878389, 5.883148, 3.4776294, 4.0043707, 5.699667]
  Cluster 3: [12.090454, 13.309772, 0.8570864, 5.217409, 2.900659, 3.3437502, 5.005318]
  Cluster 4: [11.984763, 13.2935705, 0.85079527, 5.2413807, 2.879738, 5.673261, 5.1219764]

Number of iterations: 10
cluster2.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [18.535925, 16.240742, 0.8817037, 6.1835194, 3.6971114, 4.731371, 6.0322595]
  Cluster 1: [12.090454, 13.309772, 0.8570864, 5.217409, 2.900659, 3.3437502, 5.005318]
  Cluster 2: [11.984763, 13.2935705, 0.85079527, 5.2413807, 2.879738, 5.673261, 5.1219764]
  Cluster 3: [14.930315, 14.592857, 0.8805428, 5.611794, 3.3118417, 2.7273207, 5.242952]
  Cluster 4: [18.86941, 16.342358, 0.8877735, 6.229118, 3.7429707, 2.7080002, 6.092971]

Number of iterations: 13
cluster2.csv test using Norm 2 on random starting points for K = 5:
  Cluster 0: [14.662652, 14.460818, 0.88074905, 5.570307, 3.2810616, 2.3630855, 5.1516933]
  Cluster 1: [16.47852, 15.347777, 0.8787964, 5.8680744, 3.4761477, 4.064778, 5.696185]
  Cluster 2: [19.151043, 16.469166, 0.88708967, 6.2688546, 3.7729375, 3.4604177, 6.1272507]
  Cluster 3: [12.090454, 13.309772, 0.8570864, 5.217409, 2.900659, 3.3437502, 5.005318]
  Cluster 4: [11.984763, 13.2935705, 0.85079527, 5.2413807, 2.879738, 5.673261, 5.1219764]

Number of iterations: 9
cluster2.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [11.709599, 13.1666, 0.848268, 5.1995, 2.8259602, 4.3486996, 5.0705204]
  Cluster 1: [18.907646, 16.370296, 0.8865558, 6.244471, 3.7396762, 2.7471476, 6.1150885]
  Cluster 2: [15.646338, 14.9351225, 0.8811854, 5.731415, 3.3937802, 3.0512948, 5.4189506]
  Cluster 3: [18.7513, 16.32435, 0.8829087, 6.1997833, 3.727652, 4.9087825, 6.0470004]
  Cluster 4: [12.3654995, 13.437001, 0.859035, 5.2724, 2.9605997, 6.3741, 5.1237]
  Cluster 5: [13.560716, 13.952143, 0.8746952, 5.3969994, 3.1373096, 2.3980265, 5.0124755]

Number of iterations: 18
cluster2.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [11.914392, 13.261516, 0.8502426, 5.2312274, 2.8639398, 5.0682874, 5.105348]
  Cluster 1: [13.391087, 13.865217, 0.8742455, 5.365999, 3.1142824, 2.4158065, 4.990673]
  Cluster 2: [19.503332, 16.611334, 0.8876733, 6.2945333, 3.8231337, 5.160066, 6.122333]
  Cluster 3: [15.719087, 14.976592, 0.88038635, 5.7451825, 3.3985002, 3.1870477, 5.4483857]
  Cluster 4: [19.385834, 16.619999, 0.8818583, 6.376333, 3.753833, 2.2241666, 6.216833]
  Cluster 5: [18.474817, 16.178148, 0.8868149, 6.154556, 3.7075183, 3.2254446, 6.0372224]

Number of iterations: 14
cluster2.csv test using Norm 2 on random starting points for K = 6:
  Cluster 0: [16.562, 15.391599, 0.8782441, 5.88816, 3.4807997, 4.1094804, 5.7251997]
  Cluster 1: [12.427368, 13.45737, 0.8608105, 5.2735786, 2.9756312, 6.4036317, 5.1186843]
  Cluster 2: [11.6741295, 13.168043, 0.84549135, 5.210175, 2.813326, 4.5365434, 5.1002183]
  Cluster 3: [14.861134, 14.558863, 0.8808136, 5.597773, 3.306159, 2.3817978, 5.192045]
  Cluster 4: [19.151043, 16.469166, 0.88708967, 6.2688546, 3.7729375, 3.4604177, 6.1272507]
  Cluster 5: [12.773573, 13.576073, 0.87033933, 5.275536, 3.0273216, 2.6091468, 4.933643]

Number of iterations: 10
cluster2.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [18.072252, 16.010126, 0.8845026, 6.103112, 3.6561623, 3.4683769, 5.932288]
  Cluster 1: [12.863077, 13.666461, 0.86268836, 5.3364835, 3.0139542, 3.8428621, 5.0854764]

Number of iterations: 8
cluster2.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [12.863077, 13.666461, 0.86268836, 5.3364835, 3.0139542, 3.8428621, 5.0854764]
  Cluster 1: [18.072252, 16.010126, 0.8845026, 6.103112, 3.6561623, 3.4683769, 5.932288]

Number of iterations: 5
cluster2.csv test using Norm infinity on random starting points for K = 2:
  Cluster 0: [12.962371, 13.71548, 0.8629399, 5.3548512, 3.0259707, 3.8372748, 5.1062665]
  Cluster 1: [18.240803, 16.078133, 0.88550407, 6.121159, 3.6773462, 3.453468, 5.9513206]

Number of iterations: 11
cluster2.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [14.54361, 14.407083, 0.8792875, 5.5422645, 3.2667506, 2.6127667, 5.1766667]
  Cluster 1: [18.641909, 16.26206, 0.8850365, 6.197968, 3.7139995, 3.6103494, 6.0483336]
  Cluster 2: [11.951999, 13.275066, 0.8512495, 5.2330265, 2.8682535, 4.8196125, 5.0924]

Number of iterations: 10
cluster2.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [11.951999, 13.275066, 0.8512495, 5.2330265, 2.8682535, 4.8196125, 5.0924]
  Cluster 1: [14.54361, 14.407083, 0.8792875, 5.5422645, 3.2667506, 2.6127667, 5.1766667]
  Cluster 2: [18.641909, 16.26206, 0.8850365, 6.197968, 3.7139995, 3.6103494, 6.0483336]

Number of iterations: 10
cluster2.csv test using Norm infinity on random starting points for K = 3:
  Cluster 0: [18.641909, 16.26206, 0.8850365, 6.197968, 3.7139995, 3.6103494, 6.0483336]
  Cluster 1: [11.951999, 13.275066, 0.8512495, 5.2330265, 2.8682535, 4.8196125, 5.0924]
  Cluster 2: [14.54361, 14.407083, 0.8792875, 5.5422645, 3.2667506, 2.6127667, 5.1766667]

Number of iterations: 13
cluster2.csv test using Norm infinity on random starting points for K = 4:
  Cluster 0: [18.674358, 16.274998, 0.8852033, 6.200306, 3.7180157, 3.6356776, 6.0510974]
  Cluster 1: [14.927375, 14.595247, 0.88001966, 5.6136394, 3.31023, 2.6650527, 5.2393117]
  Cluster 2: [11.900002, 13.268001, 0.848255, 5.2424746, 2.86095, 5.709325, 5.131275]
  Cluster 3: [12.204253, 13.348296, 0.8599086, 5.2221704, 2.9239998, 3.418915, 5.0144258]

Number of iterations: 13
cluster2.csv test using Norm infinity on random starting points for K = 4:
  Cluster 0: [18.674358, 16.274998, 0.8852033, 6.200306, 3.7180157, 3.6356776, 6.0510974]
  Cluster 1: [14.927375, 14.595247, 0.88001966, 5.6136394, 3.31023, 2.6650527, 5.2393117]
  Cluster 2: [11.900002, 13.268001, 0.848255, 5.2424746, 2.86095, 5.709325, 5.131275]
  Cluster 3: [12.204253, 13.348296, 0.8599086, 5.2221704, 2.9239998, 3.418915, 5.0144258]

Number of iterations: 10
cluster2.csv test using Norm infinity on random starting points for K = 4:
 Cluster 0: [18.674358, 16.274998, 0.8852033, 6.200306, 3.7180157, 3.6356776, 6.0510974]
 Cluster 1: [14.927375, 14.595247, 0.88001966, 5.6136394, 3.31023, 2.6650527, 5.2393117]
 Cluster 2: [12.204253, 13.348296, 0.8599086, 5.2221704, 2.9239998, 3.418915, 5.0144258]
 Cluster 3: [11.900002, 13.268001, 0.848255, 5.2424746, 2.86095, 5.709325, 5.131275]

Number of iterations: 30
cluster2.csv test using Norm infinity on random starting points for K = 5:
 Cluster 0: [12.116135, 13.310226, 0.85870457, 5.2097044, 2.9102044, 3.4618638, 5.0051594]
 Cluster 1: [19.19182, 16.48068, 0.8876955, 6.2712955, 3.7764091, 3.2536376, 6.1329784]
 Cluster 2: [14.698182, 14.481819, 0.88014907, 5.5788364, 3.2841458, 2.4067855, 5.170599]
 Cluster 3: [11.900002, 13.268001, 0.848255, 5.2424746, 2.86095, 5.709325, 5.131275]
 Cluster 4: [16.89, 15.534446, 0.8788779, 5.9367776, 3.5196295, 4.4745927, 5.777148]

Number of iterations: 14
cluster2.csv test using Norm infinity on random starting points for K = 5:
 Cluster 0: [14.813469, 14.5414295, 0.87993675, 5.6008787, 3.297123, 2.3519223, 5.183122]
 Cluster 1: [11.8894, 13.2642, 0.8481161, 5.23882, 2.85504, 5.41404, 5.1237397]
 Cluster 2: [19.16348, 16.475218, 0.88698924, 6.2741094, 3.771826, 3.3351526, 6.1343045]
 Cluster 3: [12.303157, 13.376316, 0.8632973, 5.2237897, 2.9484475, 3.074658, 4.980553]
 Cluster 4: [16.615183, 15.39074, 0.8807482, 5.8701854, 3.4981847, 4.47563, 5.707258]

Number of iterations: 15
cluster2.csv test using Norm infinity on random starting points for K = 5:
  Cluster 0: [12.116135, 13.310226, 0.85870457, 5.2097044, 2.9102044, 3.4618638, 5.0051594]
  Cluster 1: [16.662691, 15.418845, 0.8800462, 5.885308, 3.4994612, 4.509308, 5.717576]
  Cluster 2: [14.698182, 14.481819, 0.88014907, 5.5788364, 3.2841458, 2.4067855, 5.170599]
  Cluster 3: [19.16348, 16.475218, 0.88698924, 6.2741094, 3.771826, 3.3351526, 6.1343045]
  Cluster 4: [11.838976, 13.244873, 0.8470718, 5.2385125, 2.8497436, 5.6843076, 5.1346154]

Number of iterations: 8
cluster2.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [12.612856, 13.517144, 0.86588573, 5.2882857, 3.0194287, 7.3888574, 5.0971427]
  Cluster 1: [12.003871, 13.328064, 0.8480612, 5.255323, 2.8688064, 5.395225, 5.146258]
  Cluster 2: [11.795383, 13.173333, 0.8535973, 5.187462, 2.8497694, 3.7128205, 5.033411]
  Cluster 3: [15.366569, 14.797714, 0.88129145, 5.6793714, 3.3609715, 2.17712, 5.319685]
  Cluster 4: [14.137949, 14.230766, 0.87631273, 5.4969482, 3.2135637, 3.203846, 5.122179]
  Cluster 5: [18.785425, 16.32169, 0.8855407, 6.2133718, 3.7310843, 3.595238, 6.0715933]

Number of iterations: 15
cluster2.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [11.900002, 13.268001, 0.848255, 5.2424746, 2.86095, 5.709325, 5.131275]
  Cluster 1: [12.103863, 13.314545, 0.8573887, 5.218727, 2.9029772, 3.4231596, 5.012341]
  Cluster 2: [14.658462, 14.452308, 0.8812462, 5.560769, 3.281564, 1.9897745, 5.142923]
  Cluster 3: [19.230526, 16.49579, 0.88787633, 6.2808156, 3.7799473, 3.0437639, 6.1433163]
  Cluster 4: [17.881302, 15.969566, 0.88047826, 6.090174, 3.628044, 4.528522, 5.938522]
  Cluster 5: [15.219233, 14.735001, 0.88059616, 5.655923, 3.3489995, 3.8703845, 5.357499]

Number of iterations: 11
cluster2.csv test using Norm infinity on random starting points for K = 6:
  Cluster 0: [18.954546, 16.38879, 0.8867999, 6.247485, 3.7446969, 2.7235458, 6.119455]
  Cluster 1: [11.936274, 13.2819605, 0.8490237, 5.241922, 2.8637252, 5.4389606, 5.121333]
  Cluster 2: [12.26973, 13.363514, 0.86265403, 5.220433, 2.942838, 3.087838, 4.975217]
  Cluster 3: [19.583334, 16.646, 0.88772666, 6.315867, 3.835067, 5.0815334, 6.1443996]
  Cluster 4: [14.743958, 14.503543, 0.8803792, 5.5876675, 3.2891672, 2.3249207, 5.1749163]
  Cluster 5: [16.472692, 15.343461, 0.87895393, 5.860962, 3.4765766, 4.142654, 5.689231]