Clustering Algorithms Assignment

K.N.Toosi University of Technology
Introduction to Data Mining

Fall 2024

Part I

Practical Assignment: K-means Clustering

Task

K-means clustering struggles with **differing sizes**, **densities**, and **non-globular shapes** (e.g., spirals). Apply K-means with 2 clusters to the following datasets with initial centroids "×". While it converges well in globular distributions, even with misleading centroids, it may falter with non-globular data. State final clusters and centroids, and analyze K-means performance and limitations.

Data Points

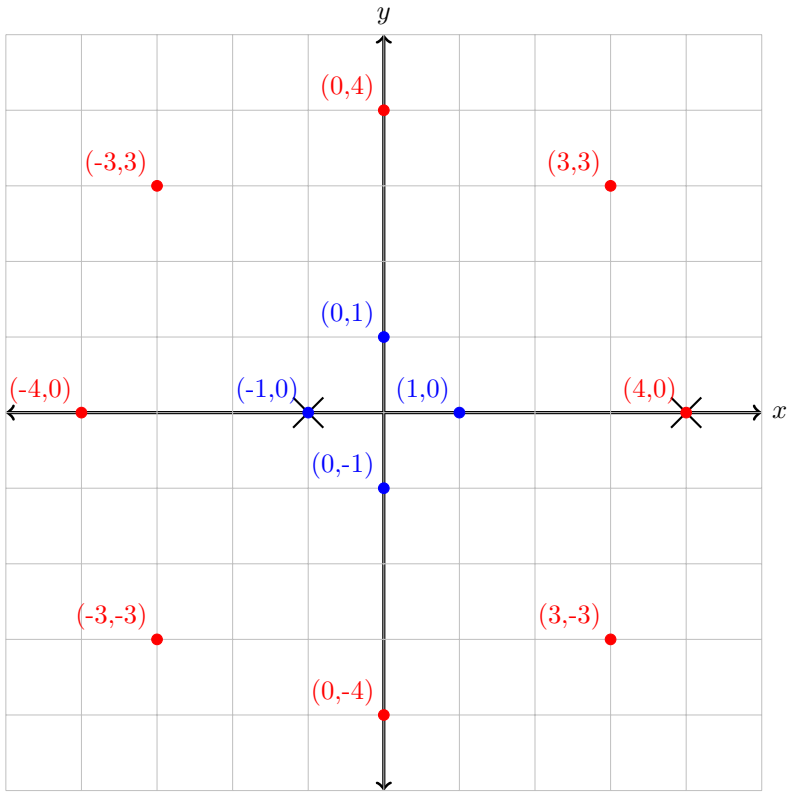

Figure 1: Non-globular Data Points

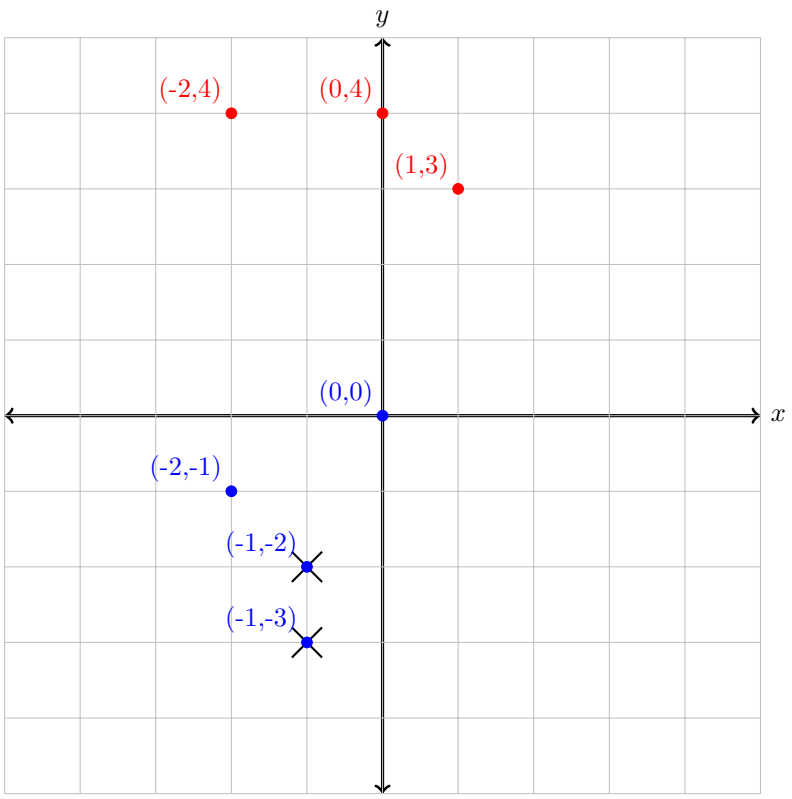

Figure 2: Globular Data Points

Part II

Practical Assignment: Agglomerative Clustering

Task

In this exercise, you'll manually perform agglomerative clustering with centroid linkage on a given set of 7 data points. Your tasks include calculating a proximity matrix, performing the clustering steps, constructing a dendrogram, and visually representing the clustering on a 2D grid. Finally, analyze the dendrogram to determine and justify the best number of clusters.

Data Points

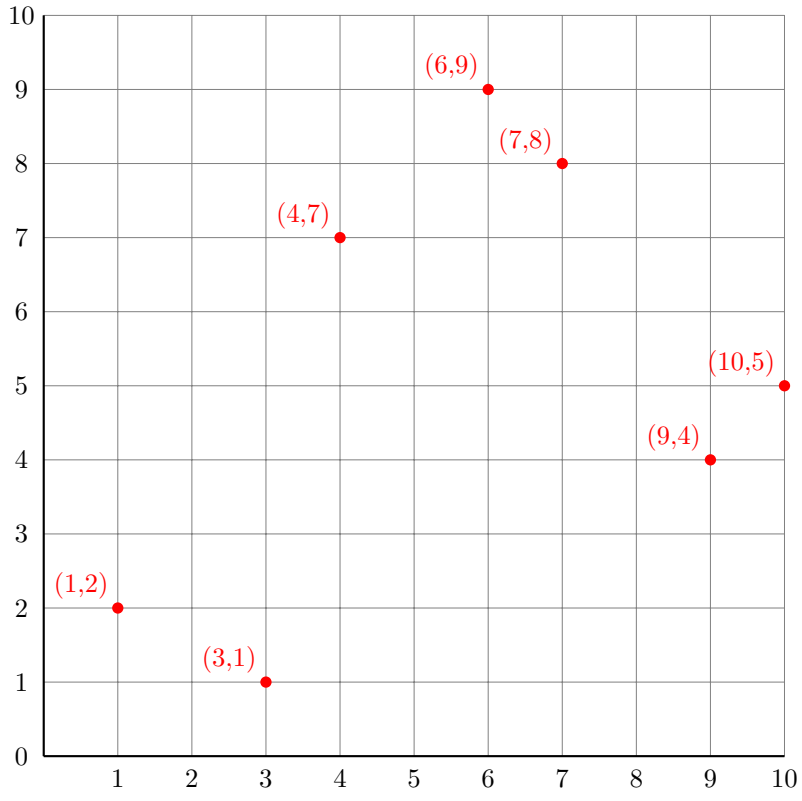

Figure 3: Agglomerative Clustering Data Points

Part III
Implementation Assignment: DBSCAN

Dataset

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(42)
n = 500
x_o = np.array([.3,-1,-1.5])
y_o = np.array([-.6,-1.5,2])
x_b = np.random.normal(-.25, .1, n//5)
y_b = np.random.normal(0, .1, n//5)
theta = np.random.uniform(0, 10, n)
r = .5 + .15 * theta
x_s = r * np.cos(theta)
y_s = r * np.sin(theta) + np.random.normal(0, .1, n)
x_1 = np.hstack([x_o, x_b, x_s])
x_2 = np.hstack([y_o, y_b, y_s])


# Dataset X
X = np.vstack([x_1,x_2]).T


plt.figure(figsize=(8, 8))
plt.scatter(X[:,0],X[:,1])
plt.grid(True)
plt.show()
```

Task

In this exercise, you will implement the DBSCAN algorithm from scratch and test it on synthesized data. Your goal is to correctly cluster the data by adjusting the DBSCAN parameters and to report the parameters that yield the best clustering results.

Part IV
Implementation Assignment: Clustering Algorithms and PCA

Dataset

Task

The goal of this assignment is to compare different clustering algorithms, K-means and DBSCAN. You will visualize the results in 2D and 3D using PCA, and determine which algorithm performs best for the given data. You will also evaluate the impact of the number of clusters on K-means and analyze DBSCAN's behavior with respect to high-dimensional data.

K-means:

1. Load the Digits Dataset:
   - Use the Digits dataset from *sklearn.datasets*.
   - Preprocess the data by standardizing it (using *standardscaler*) for better clustering performance.
2. Apply K-means clustering:
   - Perform K-means clustering for a range of k values (e.g., 1 to 20).
   - Plot the inertia (within-cluster sum of squares) for different k values and analyze the elbow method to identify the optimal number of clusters.
3. Visualize K-means clusters in 2D and 3D using PCA:
   - Reduce the data to 2D and 3D using PCA.
   - In each case, plot the projected points in two side-by-side figures: one where the projected points are colored according to the true labels (ground truth) and the other where they are colored based on the predicted labels using K-means.

DBSCAN:

1. Apply DBSCAN clustering.
2. Experiment with DBSCAN using different values for eps (epsilon) and min_samples.
3. Visualize DBSCAN clusters in 2D and 3D using PCA:

- In each case, plot the projected points in two side-by-side figures: one where the projected points are colored according to the true labels (ground truth) and the other where they are colored based on the predicted labels using DBSCAN.
- Does DBSCAN perform better than K-means or not? Explain why.

Note

Any attempt to use AI tools for generating the code is strictly prohibited. Students will be asked to present and explain their code during a class session.