Άσκηση 2.3

Ερώτημα Β

- Ο πίνακας προτύπων (pattern matrix) $D(X)$ είναι ο $N \times l$ η i-οστή γραμμή του οποίου είναι το (αντεστραμμένο) i-οστό διάνυσμα του X.
- Ο πίνακας ομοιότητας (ανομοιότητας), $P(X)$, είναι ένας $N \times N$, του οποίου το στοιχείο (i, j) ισούται με το βαθμό ομοιότητας $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (ανομοιότητας $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$) των διανυσμάτων $\boldsymbol{x}_i$ και $\boldsymbol{x}_j$. Ο πίνακας αυτός είναι γνωστός και ως πίνακας εγγύτητας.

```
PS G:\My Drive\Ece Ntua\7th Semester\Machine Learning\Series of Exercises\Exercise 2> python3 .\Exercise_3b_c_d.py

D(X) - Distance Matrix:
[[1 5]
 [3 4]
 [0 2]
 [5 4]
 [2 6]
 [3 3]
 [2 3]
 [4 2]]

P(X) - Proximity Matrix:
         Point 1  Point 2  Point 3  Point 4  Point 5  Point 6  Point 7  Point 8
Point 1    0.000    0.098    0.019    0.234    0.008    0.168    0.075    0.386
Point 2    0.098    0.000    0.200    0.032    0.051    0.010    0.002    0.106
Point 3    0.019    0.200    0.000    0.375    0.051    0.293    0.168    0.553
Point 4    0.234    0.032    0.375    0.000    0.160    0.006    0.047    0.022
Point 5    0.008    0.051    0.051    0.160    0.000    0.106    0.035    0.293
Point 6    0.168    0.010    0.293    0.006    0.106    0.000    0.019    0.051
Point 7    0.075    0.002    0.168    0.047    0.035    0.019    0.000    0.132
Point 8    0.386    0.106    0.553    0.022    0.293    0.051    0.132    0.000
```

Βοηθητικός κώδικας:

```python
import numpy as np

import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, dendrogram
from scipy.spatial.distance import pdist, squareform

# Data
X = np.array([[1,5],[3,4],[0, 2], [5, 4], [2, 6], [3, 3], [2, 3], [4, 2]])

# D(X)
D = X

# Function to compute the proximity matrix using dc(x,y)=1 - cos(theta_xy)
def proximity_matrix(X):
    n = X.shape[0]
    P = np.zeros((n, n))  # Initialize proximity matrix
    for i in range(n):
        for j in range(n):
            if i != j:
                # Compute cosine similarity and transform to proximity
```

```
                cos_theta = np.dot(X[i], X[j]) / (np.linalg.norm(X[i]) *
np.linalg.norm(X[j]))
                P[i, j] = 1 - cos_theta
    return P

# P(X)
P = proximity_matrix(X)

# Convert matrices to pandas DataFrames for better formatting

P_df = pd.DataFrame(P, index=[f"Point {i+1}" for i in range(X.shape[0])],
                    columns=[f"Point {i+1}" for i in range(X.shape[0])])

# Print the matrices in a nicely formatted way
print("\nD(X) - Distance Matrix:")
print(D)

print("\nP(X) - Proximity Matrix:")
print(P_df.round(3))  # Round to 3 decimal places for better readability
```

Ερώτημα Γ

Αρχικό Στάδιο

Ξεκινάμε με 8 μονομελή συμπλέγματα:

{1},{2},{3},{4},{5},{6},{7},{8}.

Αρχικά, εντοπίζουμε την ελάχιστη τιμή εγγύτητας στον πίνακα $P(xi, xj)$. Η ελάχιστη εγγύτητα είναι $P(2,7) = P(7,2) = 0.002$. Συνεπώς, τα σημεία 2 και 7 ενώνονται σε ένα νέο σύμπλεγμα:

$$\{2,7\}$$

Αναζητούμε ξανά τη μέγιστη τιμή εγγύτητας. Εντοπίζουμε ότι η ελάχιστη τιμή είναι $P(4, 6) = 0.006$. Συνεπώς, τα σημεία 4 και 6 ενώνονται σε ένα νέο σύμπλεγμα:

$$\{4,6\}$$

Η επόμενη ελάχιστη τιμή στον πίνακα εγγύτητας είναι $P(1, 5) = 0.008$.. Συνεπώς, τα σημεία 1 και 5 ενώνονται:

$$\{1,5\}$$

Τώρα έχουμε τρία διμελή συμπλέγματα: {2,7},{4,6},{1,5} και δύο μονομελή: {3},{8}. Ελέγχουμε τις νέες εγγύτητες και βρίσκουμε την ελάχιστη:

$$P(\{2,7\},\{4,6\}) = \min P(x_2, x_4), P(x_2, x_6), P(x_7, x_4), P(x_7, x_6) = 0.010$$

Τα συμπλέγματα {2,7} και {4,6} ενώνονται σε ένα νέο σύμπλεγμα:

$$\{2,4,6,7\}$$

Τα συμπλέγματα είναι πλέον: {1,5},{2,4,6,7},{3},{8}. Η ελάχιστη εγγύτητα τώρα βρίσκεται μεταξύ {1,5} και {3}:

$$P(\{1,5\},\{3\}) = \min P(x_1, x_3), P(x_5, x_3) = 0.019$$

Συνεπώς, ενώνονται:

$$\{1,3,5\}$$

Τα συμπλέγματα είναι τώρα: {1,3,5},{2,4,6,7},{8}. Εντοπίζουμε τη ελάχιστη εγγύτητα μεταξύ {2,4,6,7}και {8}:

$$P(\{2,4,6,7\},\{8\}) = \min P(x_2, x_8), P(x_4, x_8), P(x_6, x_8), P(x_7, x_8) = 0.022$$

Συνεπώς, ενώνονται:

$$\{2,4,6,7,8\}$$

Απομένουν δύο συμπλέγματα: {1,3,5} και {2,4,6,7,8}. Η ελάχιστη εγγύτητα είναι:

$P(\{2,4,6,7,8\},\{1,3,5\})$
$= \min P(x_2, x_1), P(x_2, x_3), P(x_2, x_5), P(x_4, x_1), P(x_4, x_3), P(x_4, x_5), P(x_6, x_1), P(x_6, x_3), P(x_6, x_5) \ldots,$
$= 0.035$

Όλα τα σημεία ενώνονται σε ένα τελικό σύμπλεγμα:

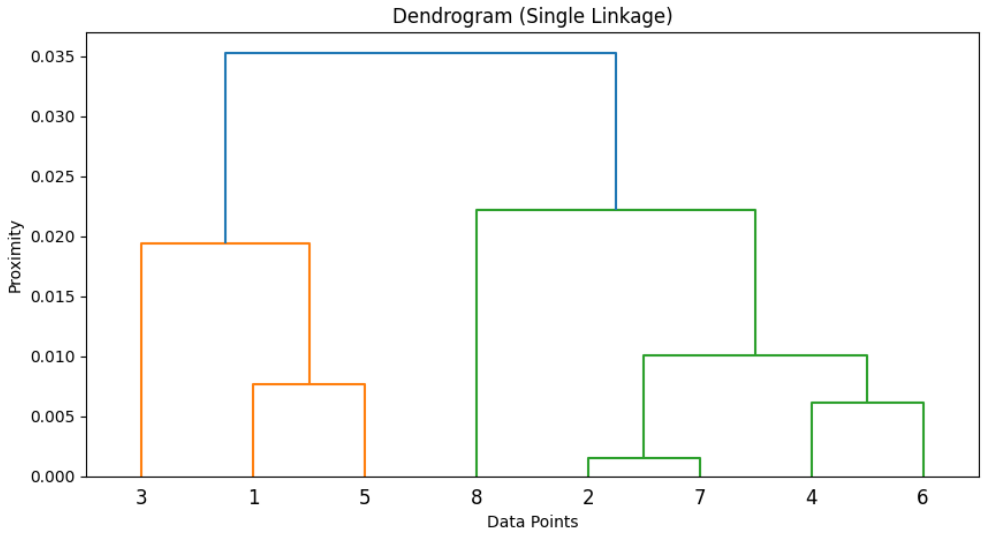
$$\{1,2,3,4,5,6,7,8\}$$

Παρακάτω δίνεται ο βοηθητικός κώδικας (έτοιμη βιβλιοθήκη χρησιμοποιήθηκε) και το αντίστοιχο δενδρόγραμμα.

```
# Convert the proximity matrix to a condensed form
P_condensed = squareform(P)

# Hierarchical Clustering with Single Linkage
linkage_matrix_single = linkage(P_condensed, method='single')
```

```
# Plot dendrogram for Single Linkage
plt.figure(figsize=(10, 5))
dendrogram(linkage_matrix_single, labels=np.arange(1, X.shape[0] + 1))
plt.title('Dendrogram (Single Linkage)')
plt.xlabel('Data Points')
plt.ylabel('Proximity')
plt.show()
```



Ερώτημα Δ

Αρχικό Στάδιο

Ξεκινάμε με 8 μονομελή συμπλέγματα:

{1},{2},{3},{4},{5},{6},{7},{8}.

Αρχικά, εντοπίζουμε την ελάχιστη τιμή εγγύτητας στον πίνακα $P(xi, xj)$. Η ελάχιστη εγγύτητα είναι $P(2,7) = P(7,2) = 0.002$. Συνεπώς, τα σημεία 2 και 7 ενώνονται σε ένα νέο σύμπλεγμα:

{2,7}

Αναζητούμε ξανά τη μέγιστη τιμή εγγύτητας. Εντοπίζουμε ότι η ελάχιστη τιμή είναι $P(4, 6) = 0.006$. Συνεπώς, τα σημεία 4 και 6 ενώνονται σε ένα νέο σύμπλεγμα:

{4,6}

Η επόμενη ελάχιστη τιμή στον πίνακα εγγύτητας είναι $P(1,5) = 0.008$. Συνεπώς, τα σημεία 1 και 5 ενώνονται:

$$\{1,5\}$$

Τώρα έχουμε τρία διμελή συμπλέγματα: $\{2,7\},\{4,6\},\{1,5\}$ και δύο μονομελή: $\{3\},\{8\}$. Ελέγχουμε τις νέες εγγύτητες και βρίσκουμε την μέγιστη:

$$P(\{2,7\},\{4,6\}) = \max P(x_2,x_4), P(x_2,x_6), P(x_7,x_4), P(x_7,x_6) = 0.047$$

Τα συμπλέγματα $\{2,7\}$ και $\{4,6\}$ ενώνονται σε ένα νέο σύμπλεγμα:

$$\{2,4,6,7\}$$

Τα συμπλέγματα είναι πλέον: $\{1,5\},\{2,4,6,7\},\{3\},\{8\}$. Η μέγιστη εγγύτητα τώρα μεταξύ $\{1,5\}$ και $\{3\}$:

$$P(\{1,5\},\{3\}) = \max P(x_1,x_3), P(x_5,x_3) = 0.051$$

Συνεπώς, ενώνονται:

$$\{1,3,5\}$$

Τα συμπλέγματα είναι τώρα: $\{1,3,5\},\{2,4,6,7\},\{8\}$. Εντοπίζουμε τη μέγιστη εγγύτητα μεταξύ $\{2,4,6,7\}$και $\{8\}$:

$$P(\{2,4,6,7\},\{8\}) = max\, P(x_2,x_8), P(x_4,x_8), P(x_6,x_8), P(x_7,x_8) = 0.132$$

Συνεπώς, ενώνονται:

$$\{2,4,6,7,8\}$$

Απομένουν δύο συμπλέγματα: $\{1,3,5\}$ και $\{2,4,6,7,8\}$. Η ελάχιστη εγγύτητα είναι:

$P(\{2,4,6,7,8\},\{1,3,5\})$
$= \max P(x_2,x_1), P(x_2,x_3), P(x_2,x_5), P(x_4,x_1), P(x_4,x_3), P(x_4,x_5)\,, P(x_6,x_1), P(x_6,x_3), P(x_6,x_5) \ldots,$
$= 0.553$

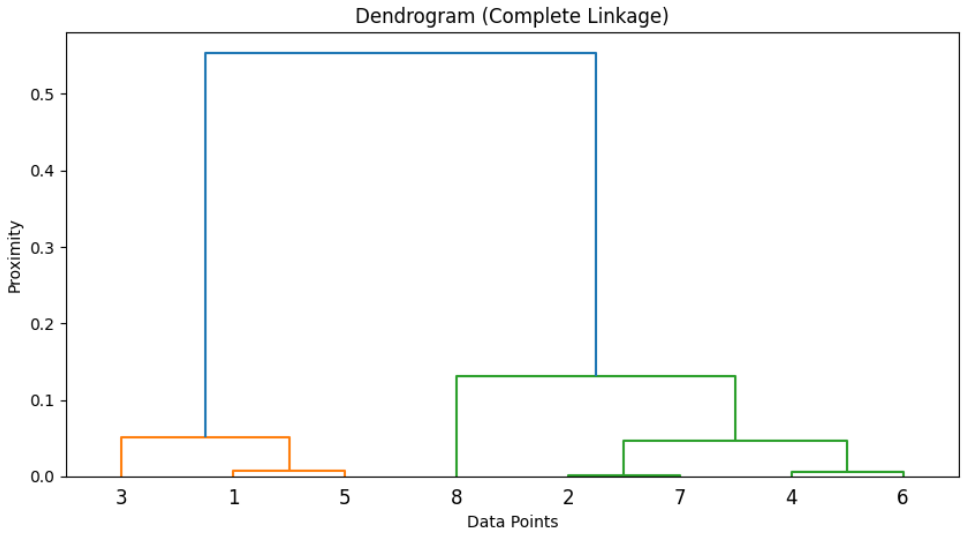
Όλα τα σημεία ενώνονται σε ένα τελικό σύμπλεγμα:

$$\{1,2,3,4,5,6,7,8\}$$

Παρακάτω δίνεται ο βοηθητικός κώδικας (έτοιμη βιβλιοθήκη χρησιμοποιήθηκε) και το αντίστοιχο δενδρόγραμμα.

```
# Hierarchical Clustering with Complete Linkage
linkage_matrix_complete = linkage(P_condensed, method='complete')

# Plot dendrogram for Complete Linkage
plt.figure(figsize=(10, 5))
dendrogram(linkage_matrix_complete, labels=np.arange(1, X.shape[0] + 1))
plt.title('Dendrogram (Complete Linkage)')
plt.xlabel('Data Points')
plt.ylabel('Proximity')
plt.show()
```



Ερώτημα Ε

Παρατηρούμε ότι τα δύο δενδρογράμματα που προκύπτουν από τις μεθόδους απλού και πλήρους δεσμού είναι πανομοιότυπα. Και στις δύο περιπτώσεις, οι ομαδοποιήσεις που προκύπτουν είναι οι εξής: ένα cluster για τα σημεία {3,1,5} και ένα για τα σημεία {2,7,4,6,8}. Το γεγονός ότι οι δύο μέθοδοι οδηγούν στο ίδιο αποτέλεσμα μπορεί να αποδοθεί στη φύση των δεδομένων. Πιο συγκεκριμένα, οι αποστάσεις μεταξύ των σημείων πρέπει να είναι τέτοιες ώστε η ελάχιστη και η μέγιστη απόσταση εντός των ομάδων να είναι παρόμοιες, ώστε τόσο ο απλός όσο και ο πλήρης δεσμός να καταλήγουν σε παρόμοια ή ταυτόσημα αποτελέσματα.