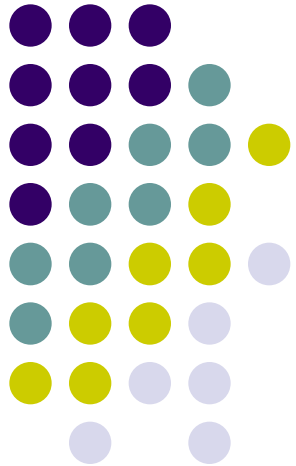
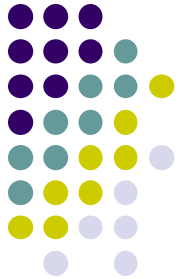
CLUSTERING
ΟΜΑΔΟΠΟΙΗΣΗ



ΕΙΣΑΓΩΓΗ-
Τι είναι ομαδοποίηση



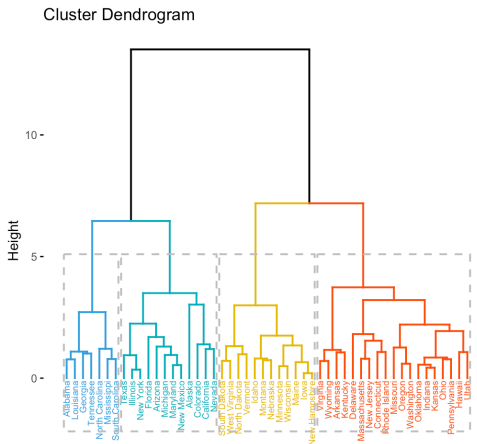
- **Ομαδοποίηση** είναι η κατηγοριοποίηση αντικειμένων σε διαφορετικές ομάδες, ή για την ακρίβεια, ο διαμερισμός ενός συνόλου δεδομένων σε υποσύνολα (ομάδες ή συστάδες), έτσι ώστε τα δεδομένα σε κάθε υποσύνολο (ιδανικά) να μοιράζονται κάποια κοινά χαρακτηριστικά- συχνά σύμφωνα με κάποιο καθορισμένο μέτρο απόστασης.

Τύποι Ομαδοποίησης:

1. **Ιεραρχικοί αλγόριθμοι**: βρίσκουν διαδοχικές ομάδες χρησιμοποιώντας ομάδες αναγνωρισμένες από πριν.

   1. Συσσωμάτωση ("bottom-up"): Οι αλγόριθμοι συσσωμάτωσης ξεκινούν με κάθε στοιχείο σαν χωριστή ομάδα και τις συνενώνουν σε αναγνωρισμένες μεγαλύτερες ομάδες.

   2. Διχαστικοί ("top-down"): Οι διχαστικοί αλγόριθμοι ξεκινούν με ολόκληρο το σύνολο και το διαιρούν σε αναγνωρισμένες μικρότερες ομάδες.

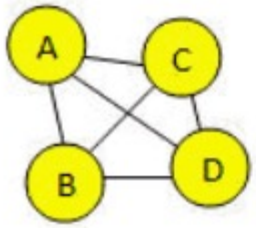


Το κριτήριο σύνδεσης

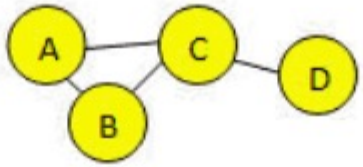
- Το κριτήριο σύνδεσης αποφασίζει πώς θα γίνει η σύνδεση ή ο διαχωρισμός των ομάδων

Maximum or complete-linkage clustering	$\max_{a \in A,\, b \in B} d(a,b)$
Minimum or single-linkage clustering	$\min_{a \in A,\, b \in B} d(a,b)$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{\lvert A\rvert \cdot \lvert B\rvert} \sum_{a \in A} \sum_{b \in B} d(a,b).$

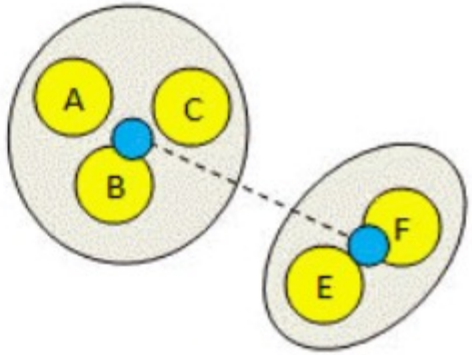
Το κριτήριο σύνδεσης



Maximum distance
Complete linkage

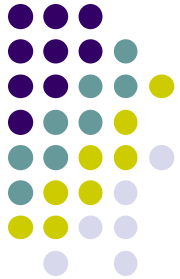


Minimum distance
Single linkage



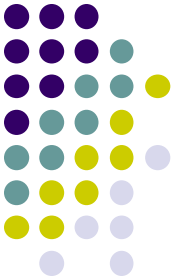
Average linkage

Τύποι Ομαδοποίησης:



2. **Ομαδοποίηση Διαμερισμού:** Οι αλγόριθμοι διαμερισμού τις καθορίζουν όλες από την αρχή. Αυτοί περιλαμβάνουν:
   - ***K*-means και παραλλαγές του**
   - Fuzzy *c*-means ομαδοποίηση
   - QT αλγόριθμος ομαδοποίησης

Common Distance measures:



- *Η μετρική απόστασης* θα καθορίσει πως υπολογίζεται η ομοιότητα δύο στοιχείων και θα επηρεάσει το σχήμα των ομάδων.

  Συνήθεις μετρικές είναι:

1. Η Manhattan απόσταση (επίσης ονομάζεται νόρμα-1) και δίνεται από:

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

2. Η Ευκλείδια απόσταση (επίσης ονομάζεται νόρμα-2) και δίνεται από:

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

3.Η maximum νόρμα δίνεται από:

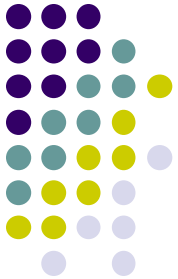
$$d(x,y) = \max_{1 \le i \le p} |x_i - y_i|$$

4. Η απόσταση Mahalanobis διορθώνει δεδομένα για διαφορετικές κλίμακες και συσχετίσεις στις μεταβλητές.

5. Εσωτερικό γινόμενο: Η γωνία μεταξύ 2 διανυσμάτων μπορεί να χρησιμοποιηθεί σαν μέτρο απόστασης, όταν ομαδοποιούνται δεδομένα μεγάλης διάστασης.

6. Απόσταση Hamming: μετρά τον ελάχιστο αριθμό αντικαταστάσεων που απαιτούνται για να γίνει αλλαγή από το ένα μέλος στο άλλο.

ΟΜΑΔΟΠΟΙΗΣΗ K-MEANS

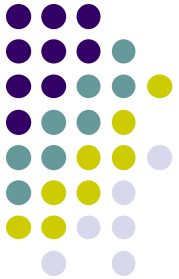


- Ο αλγόριθμος **k-means** είναι ένας αλγόριθμος για να ομαδοποιεί *n* αντικείμενα βασιζόμενος σε χαρακτηριστικά, σε *k* διαμερισμούς, όπου $k < n$.
- Είναι παρόμοιος με τον αλγόριθμο expectation-maximization για μίξη Gaussians γιατί και οι δύο προσπαθούν να βρουν τα κέντρα των φυσικών ομάδων στα δεδομένα.
- Υποθέτει ότι τα χαρακτηριστικά των δεδομένων σχηματίζουν ένα διανυσματικό χώρο.

- Είναι ένας αλγόριθμος για διαμερισμό (ή ομαδοποίηση) N σημείων δεδομένων σε *K* διαφορετικά υποσύνολα S_j που περιέχουν δεδομένα τέτοια ώστε να ελαχιστοποιούν το κριτήριο sum-of-squares.

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left| x_n - \mu_j \right|^2,$$

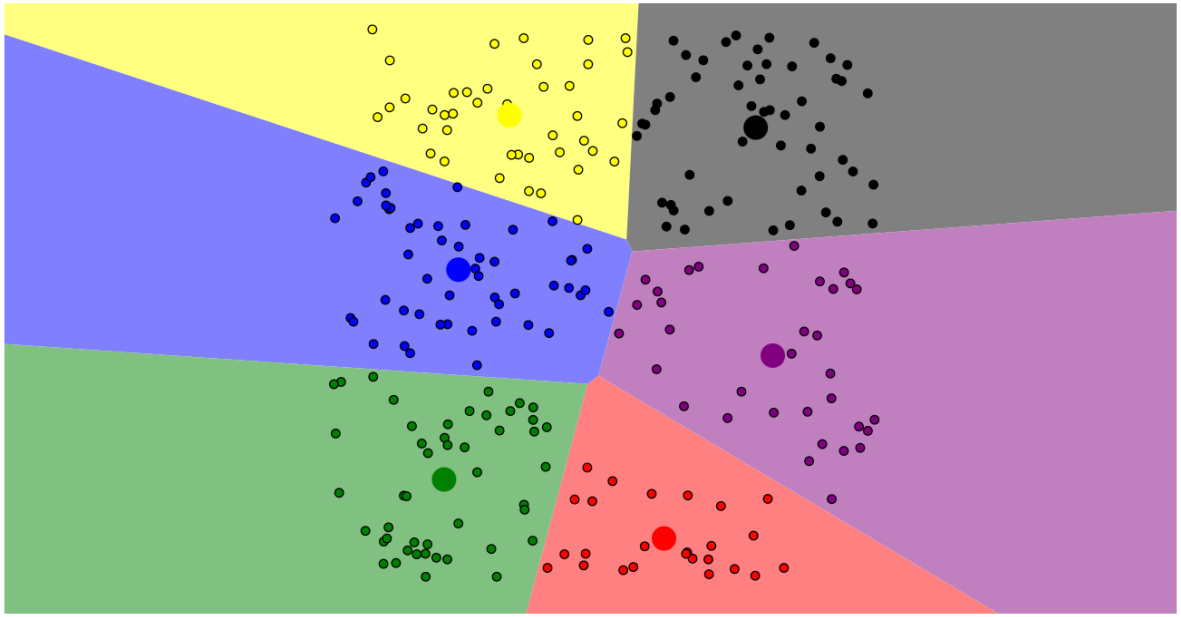
όπου x_n είναι ένα διάνυσμα που αναπαριστά το $n^{στό}$ σημείο δεδομένων και μ_j είναι το γεωμετρικό κέντρο των δεδομένων στο S_j.



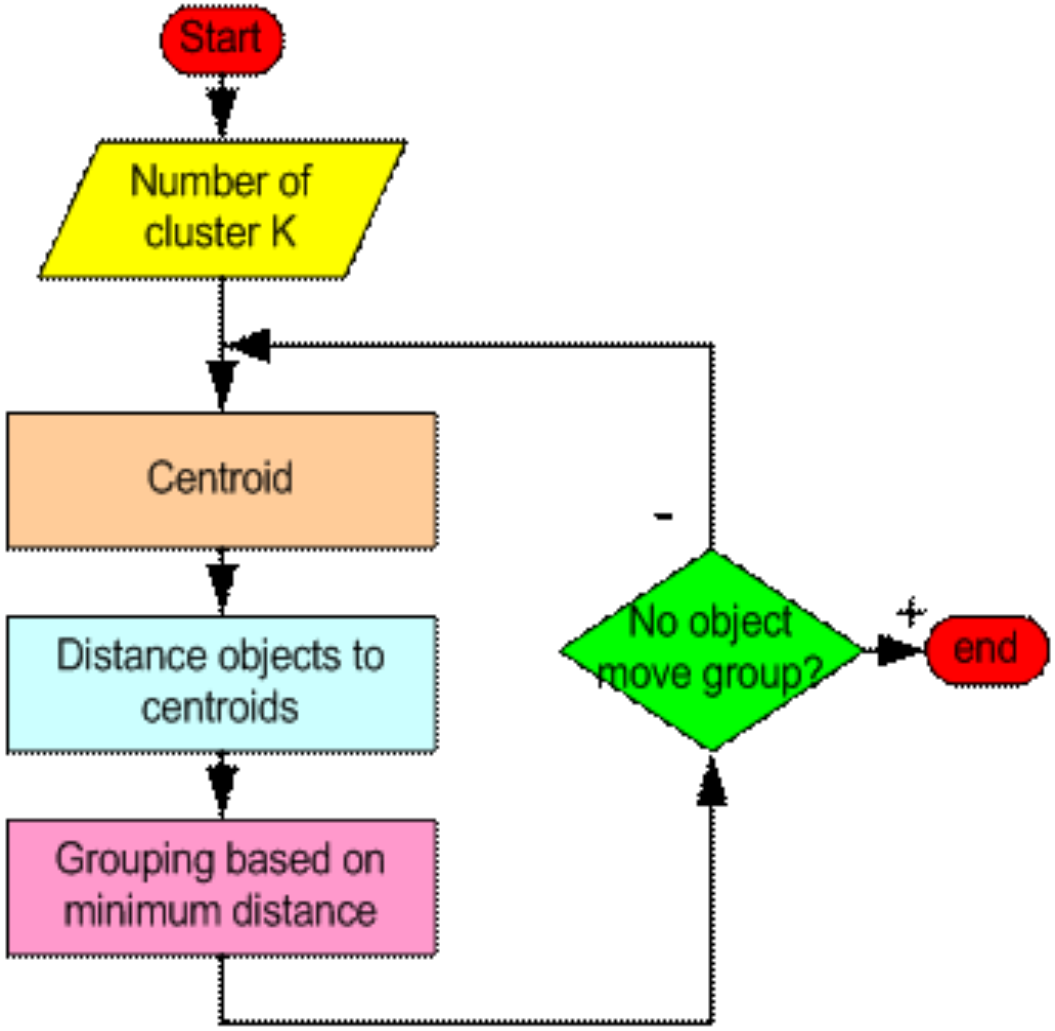
- Μιλώντας απλά η k-means ομαδοποίηση είναι ένας αλγόριθμος που κατηγοριοποιεί ή ομαδοποιεί τα αντικείμενα, βασιζόμενη σε χαρακτηριστικά, σε *K* ομάδες.
- Το *K* είναι θετικός ακέραιος αριθμός.
- Η ομαδοποίηση γίνεται ελαχιστοποιώντας το άθροισμα τετραγώνων των αποστάσεων μεταξύ των δεδομένων και του αντίστοιχου κέντρου της ομάδας.

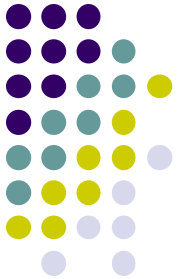
Επίδειξη

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/



Πως λειτουργεί ο αλγόριθμος ομαδοποίησης K-Means;





- **Step 1:** Αρχικά αποφασίζουμετην τιμή του k = ο αριθμός των ομάδων.
- **Step 2**: Κάνε ένα αρχικό διαμερισμό, που κατηγοριοποιεί τα δεδομένα σε k ομάδες. Μπορείς να αντιστοιχίσεις τα εκπαιδευτικά δείγματα τυχαία ή συστηματικά ως εξής:
    1. Πάρε τα πρώτα k πρότυπα σαν ομάδες ενός δείγματος.
    2. Αντιστοίχισε κάθε ένα από τα υπόλοιπα δείγματα στην ομάδα με το πλησιέστερο κέντρο. Μετά, υπολόγισε πάλι το κέντρο της νικήτριας ομάδας.

- **Step 3:** Πάρε κάθε ένα δείγμα και υπολόγισε την απόσταση από το κέντρο κάθε ομάδας. Αν ένα τρέχον δείγμα δεν είναι στην ομάδα με το πλησιέστερο κέντρο, μεταφέρεται σε αυτή την ομάδα και ενημερώνεται το κέντρο της ομάδας που κέρδισε το νέο δείγμα και της ομάδας που έχασε το δείγμα.
- **Step 4 .** Επανάλαβε το step 3 μέχρι να επιτύχεις σύγκλιση, δηλαδή μέχρι ένα πέρασμα όλου του εκπαιδευτικού συνόλου, δεν οδηγεί σε νέες αντιστοιχίσεις.

Ένα απλό παράδειγμα υλοποίησης του αλγορίθμου k-means (K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Αρχικοποίηση: Επιλέγουμε τυχαία τα κέντρα (k=2) για 2 ομάδες.

Άρα τα 2 κέντρα είναι: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Έτσι παίρνουμε 2 ομάδες: {1,2,3} {4,5,6,7}.
- Τα νέα κέντρα τους είναι:

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Τώρα, υπολογίζουμε την Ευκλείδεια απόσταση κάθε δείγματος χρησιμοποιώντας αυτά τα κέντρα, όπως φαίνεται στον πίνακα.

- Έτσι, οι νέες ομάδες είναι:
  {1,2} {**3**,4,5,6,7}

- Τα επόμενα κέντρα είναι: m1=(1.25,1.5) , m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

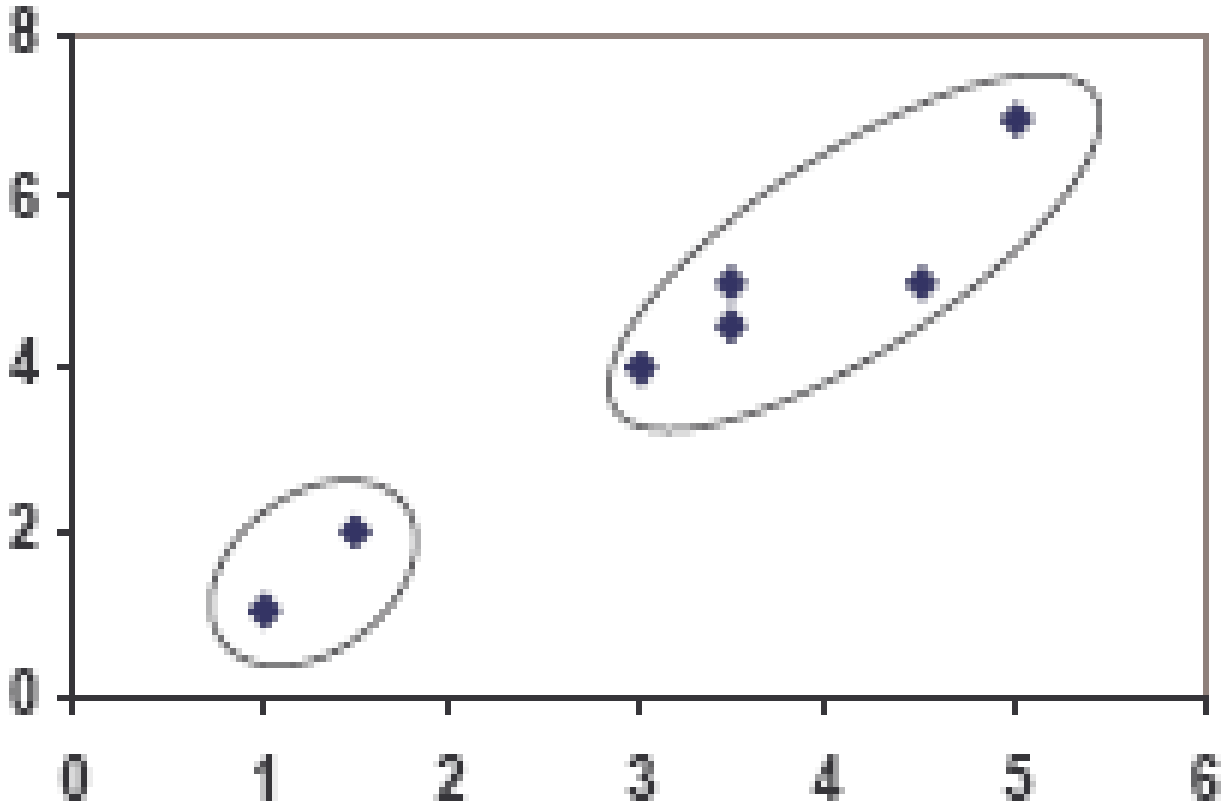
  Οι ομάδες που παίρνουμε είναι:

  {1,2} and {3,4,5,6,7}

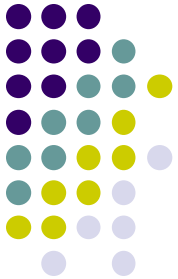
- Άρα, δεν υπάρχει καμία αλλαγή στις ομάδες.
- Έτσι, ο αλγόριθμος σταματά εδώ και το τελικό αποτέλεσμα είναι 2 ομάδες {1,2} και {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



(με K=3)



Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

C_3

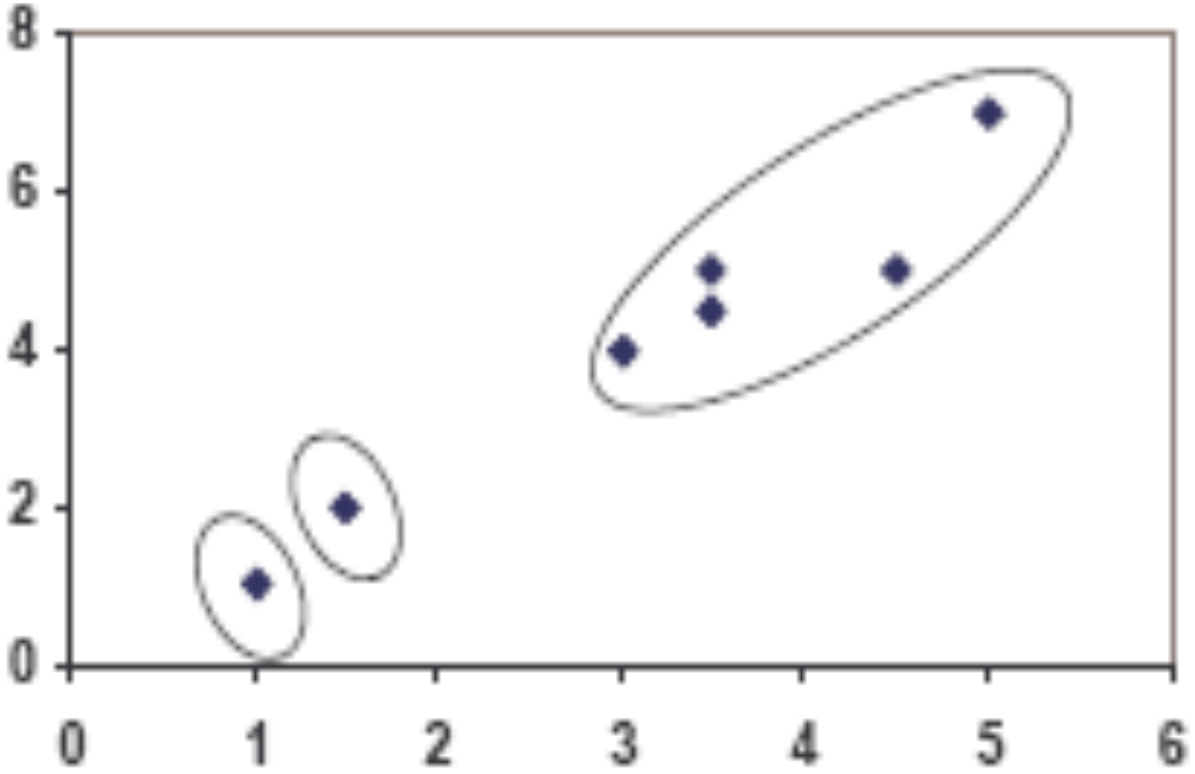
clustering with initial centroids (1, 2, 3)

Step 1

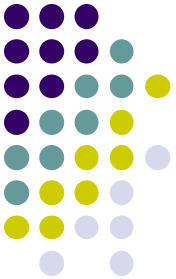
Individual	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9,5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

Step 2

PLOT



Αριθμητικό Παράδειγμα Ομαδοποίησης K-Means

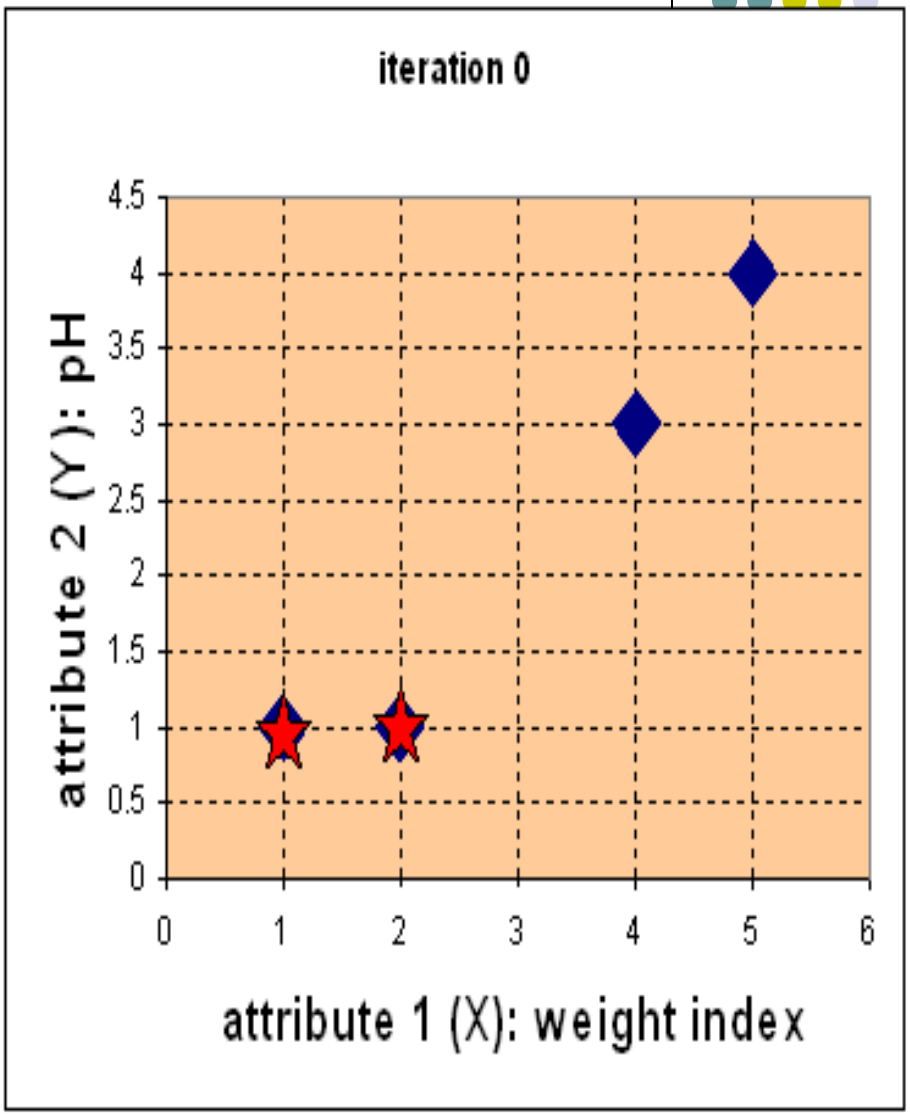


Έχουμε 4 φάρμακα σαν δείγματα εκπαίδευσης και κάθε φάρμακο έχει 2 χαρακτηριστικά. Κάθε χαρακτηριστικό αναπαριστά συνιστώσα του δείγματος. Πρέπει να καθορίσουμε ποια φάρμακα ανήκουν στην ομάδα 1 και ποια στην άλλη ομάδα.

Object	Attribute1 (X): weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Step 1:

- Αρχική τιμή των κέντρων: Υποθέστε ότι χρησιμοποιούμε τα φάρμακα A και B σαν τα αρχικά κέντρα.
- Έστω ότι c_1 και c_2 οι συντεταγμένες των κέντρων, τότε c_1=(1,1) και c_2=(2,1)



- **<u>Απόσταση κέντρων - δειγμάτων</u>** : Υπολογίζουμε την απόσταση κάθε δείγματος από το κέντρο των ομάδων. Έστω ότι χρησιμοποιούμε την Euclidean distance, τότε στην επανάληψη 0 έχουμε τον πίνακα απόστασης:

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

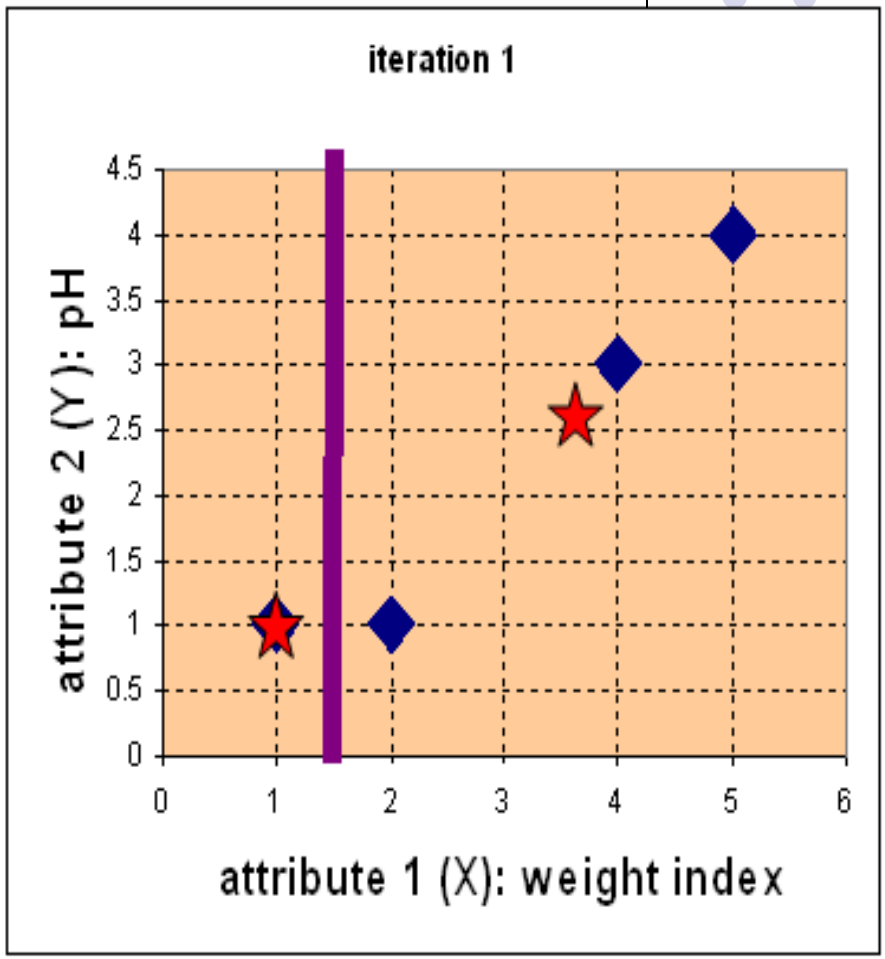
- Κάθε στήλη, στον πίνακα απόστασης αναπαριστά το δείγμα.
- Η 1η γραμμή στον πίνακα αντιστοιχεί στην απόσταση κάθε δείγματος από το 1ο κέντρο και η 2η γραμμή είναι η απόσταση κάθε δείγματος από το 2ο κέντρο.
- Για παράδειγμα, η απόσταση του C = (4, 3) από το 1ο κέντρο $\mathbf{c}_1 = (1,1)$ είναι: $\sqrt{(4-1)^2 + (3-1)^2} = 3.61$ και η απόσταση από το 2ο κέντρο, $\mathbf{c}_2 = (2,1)$ είναι $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$, κλπ.

Step 2:

- **Ομαδοποίηση δειγμάτων**: Αντιστοιχίζουμε κάθε δείγμα, με βάση την ελάχιστη απόσταση.
- Το φάρμακο A αντιστοιχίζεται στην ομάδα 1, το B στην ομάδα 2, το C στη 2 και το D στη 2.
- Τα στοιχεία του Πίνακα Ομάδας παρακάτω είναι 1 αν και μόνο αν το δείγμα αντιστοιχίζεται στην ομάδα.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$A \quad B \quad C \quad D$$



- **<u>Επανάληψη-1, Απόσταση δειγμάτων- κέντρων</u>**: Το επόμενο βήμα είναι να υπολογίσουμε την απόσταση όλων των δειγμάτων από τα νέα κέντρα.
- Όμοια στο βήμα 2, έχουμε distance matrix:

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

- **Επανάληψη-1, Ομαδοποίηση δειγμάτων:** Βασιζόμενοι στο νέο distance matrix, μετακινούμε το Β στην ομάδα1, ενώ όλα τα άλλα δείγματα δεν αλλάζουν. Ο Group matrix φαίνεται παρακάτω:

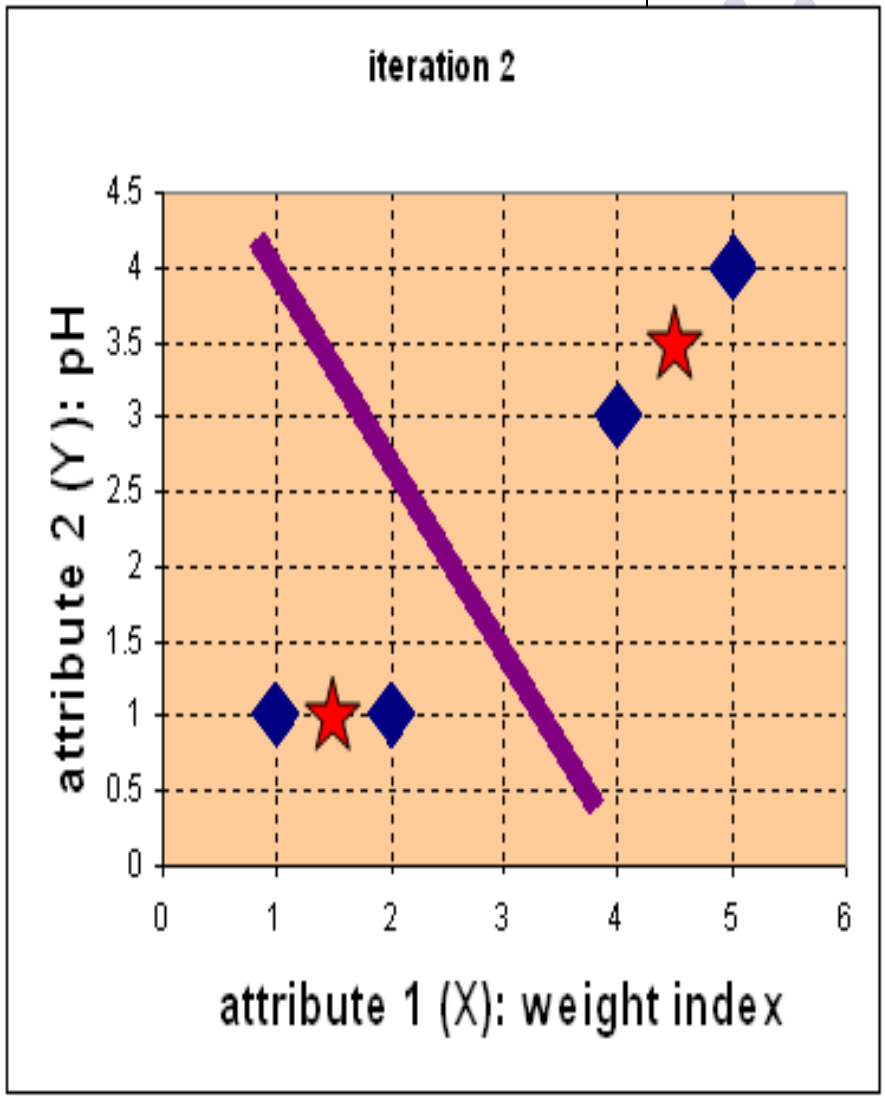
$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$A \quad B \quad C \quad D$$

- **Επανάληψη 2, καθορισμός κέντρων:** Επαναλαμβάνουμε τώρα το step 4 για να υπολογίσουμε τα νέα κέντρα βάση της ομαδοποίηση της προηγούμενης επανάληψης. Οι ομάδες 1 και 2 έχουν από 2 μέλη, έτσι τα νέα κέντρα είναι:

$$\mathbf{c}_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(1\tfrac{1}{2}, 1\right)$$

$$\mathbf{c}_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(4\tfrac{1}{2}, 3\tfrac{1}{2}\right)$$



- **<u>Επανάληψη-2, Αποστάσεις Δειγμάτων - Κέντρων</u>**: Επανάλαβε το βήμα 2. Ο νέος distance matrix, είναι:

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1\frac{1}{2}, 1) \quad group-1 \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) \quad group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

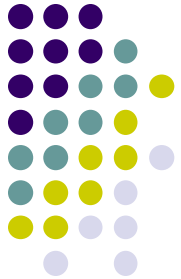
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

- **<u>Επανάληψη-2, Ομαδοποίηση Δειγμάτων:</u>** *Πάλι, αντιστοιχίζουμε κάθε δείγμα με βάση την ελάχιστη απόσταση*.

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$A \quad B \quad C \quad D \qquad\qquad \mathbf{G}^2 = \mathbf{G}^1$$

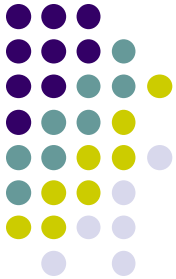
- Παίρνουμε αποτέλεσμα: . Συγκρίνοντας την ομαδοποίηση της τελευταίας επανάληψης βλέπουμε ότι τα δείγματα δεν αλλάζουν ομάδα.
- Έτσι, ο υπολογισμός της ομαδοποίησης k-mean ολοκληρώθηκε (σύγκλιση) και δεν απαιτούνται άλλοι υπολογισμοί.



Η τελική ομαδοποίηση είναι:

Object	Feature1(X): weight index	Feature2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

k-Means Clustering

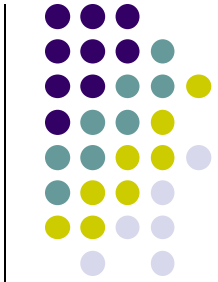


```python
from typing import List, Tuple
import numpy as np

def kmeans(X: List[List[float]], k: int, max_iter: int) -> Tuple[List[int],
List[List[float]]]:
    # Randomly initialize the centers
    centers = [X[i] for i in np.random.choice(len(X), k, replace=False)]

    for _ in range(max_iter):
        # Compute the distances between each point and the centers
        distances = [[np.linalg.norm(x - c) for c in centers] for x in X]

        # Assign each point to the closest center
        clusters = [np.argmin(d) for d in distances]

        # Update the centers as the mean of the points in each cluster
        for i in range(k):
            points = [X[j] for j in range(len(X)) if clusters[j] == i]
            centers[i] = np.mean(points, axis=0) if len(points) > 0 else
centers[i]

    return clusters, centers
```

Μειονεκτήματα της Ομαδοποίησης K-Mean

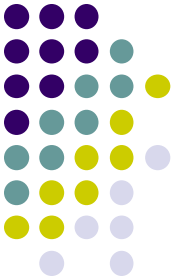
1. Ο αριθμός των ομάδων, K, πρέπει να καθοριστεί από την αρχή. Το μειονέκτημα είναι ότι δεν δίνει το ίδιο αποτέλεσμα σε κάθε τρέξιμο, γιατί οι ομάδες που προκύπτουν εξαρτώνται από τις τυχαίες αρχικές αντιστοιχίσεις.
2. Ποτέ δεν ξέρουμε την πραγματική ομαδοποίηση, χρησιμοποιώντας τα ίδια δεδομένα,τα οποία όταν εισάγονται με διαφορετική σειρά, μπορεί να παράγουν διαφορετική ομαδοποίηση, αν ο αριθμός των δεδομένων είναι μικρός.
3. Είναι ευαίσθητη στην αρχικοποίηση. Διαφορετική αρχικοποίηση ίσως παράγει διαφορετικά αποτελέσματα ομαδοποίησης. Ο αλγόριθμος ίσως παγιδευτεί σε *τοπικό βέλτιστο*.

Εφαρμογές της K-Mean ομαδοποίησης



- Είναι σχετικά αποδοτική και γρήγορη. Υπολογίζει το αποτέλεσμα σε **O**(*tkn*), όπου *n* είναι ο αριθμός των δειγμάτων, *k* είναι ο αριθμός των ομάδων και *t* ο αριθμός των επαναλήψεων.
- Η ομαδοποίηση k-means μπορεί να εφαρμοστεί σε *machine learning ή data mining*
- *Έχει χρησιμοποιηθεί σε ακουστικά δεδομένα σε κατανόηση ομιλίας για να μετατρέψει κυματοσειρές σε μια από k κατηγορίες (γνωστό σαν Vector Quantization ή Image Segmentation).*
- *Επίσης έχει χρησιμοποιηθεί για επιλογή χρωματικών παλετών σε γραφικές οθόνες.*

ΣΥΜΠΕΡΑΣΜΑ

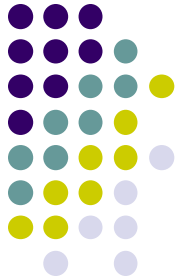


- *Ο αλγόριθμος K-means είναι χρήσιμος* για μη-επιβλεπόμενη ανακάλυψη γνώσης και είναι σχετικά απλός.
- Ο K-means έχει βρει ευρεία χρήση σε πολλά πεδία, όπως μη-επιβλεπόμενη μάθηση σε Νευρωνικά Δίκτυα, Αναγνώριση Προτύπων, Κατηγοριοποίηση, Τεχνητή Νοημοσύνη, Επεξεργασία Εικόνας, Μηχανική Όραση κ.λ.π.

References

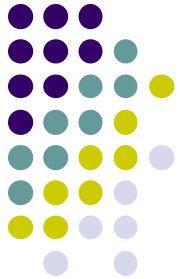
- Tutorial - Tutorial with introduction of Clustering Algorithms (k-means, fuzzy-c-means, hierarchical, mixture of gaussians) + some interactive demos (java applets).
- Digital Image Processing and Analysis-byB.Chanda and D.Dutta Majumdar.
- H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
- J. A. Hartigan (1975) "Clustering Algorithms". Wiley.
- J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.
- D. Arthur, S. Vassilvitskii (2006): "How Slow is the k-means Method?,"
- D. Arthur, S. Vassilvitskii: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).
- www.wikipedia.com

Spectral clustering (φασματική ομαδοποίηση)



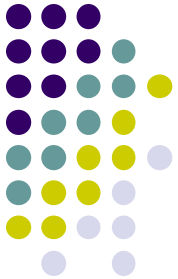
- Η φασματική ομαδοποίηση είναι μια τεχνική με ρίζες στη θεωρία γραφημάτων. Χρησιμοποιείται για τον εντοπισμό κοινοτήτων κόμβων σε ένα γράφημα.
- Η μέθοδος είναι ευέλικτη και μας επιτρέπει να ομαδοποιήσουμε και δεδομένα εκτός γραφημάτων.
- Η φασματική ομαδοποίηση χρησιμοποιεί πληροφορίες από τις ιδιοτιμές (φάσμα) ειδικών πινάκων που έχουν δημιουργηθεί από το γράφημα ή το σύνολο δεδομένων.

Ιδιοδιανύσματα και Ιδιοτιμές



- Για έναν πίνακα Α, εάν υπάρχει ένα διάνυσμα **x** που δεν είναι μηδενικό και ένα βαθμωτό μέγεθος λ τέτοιο ώστε Α**x** = λ**x**, τότε το **x** λέγεται ότι είναι ένα ιδιοδιάνυσμα του Α με αντίστοιχη ιδιοτιμή λ.
- Μπορούμε να σκεφτούμε τον πίνακα Α ως μια συνάρτηση που αντιστοιχίζει διανύσματα σε νέα διανύσματα.
- Τα περισσότερα διανύσματα θα καταλήξουν κάπου εντελώς διαφορετικά όταν το Α εφαρμόζεται σε αυτά, αλλά τα ιδιοδιανύσματα αλλάζουν μόνο ως προς το μέτρο τους.

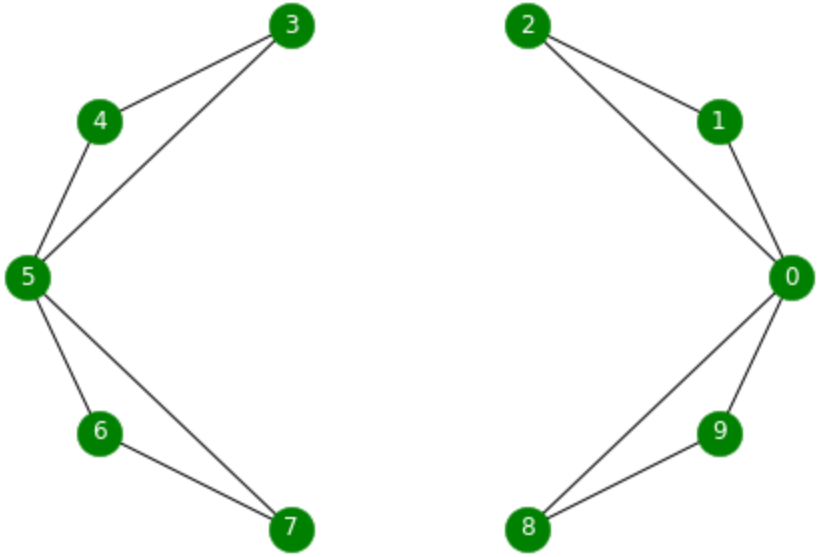
Ιδιοδιανύσματα και Ιδιοτιμές



```python
import numpy as np

# a 2x2 matrix
A = np.array([[0,1],[-2,-3]])

# find eigenvalues and eigenvectors
vals, vecs = np.linalg.eig(A)

# print results
for i, value in enumerate(vals):
    print("Eigenvector:", vecs[:,i], ", Eigenvalue:", value)

# Eigenvector: [ 0.70710678 -0.70710678] , Eigenvalue: -1.0
# Eigenvector: [-0.4472136   0.89442719] , Eigenvalue: -2.0
```

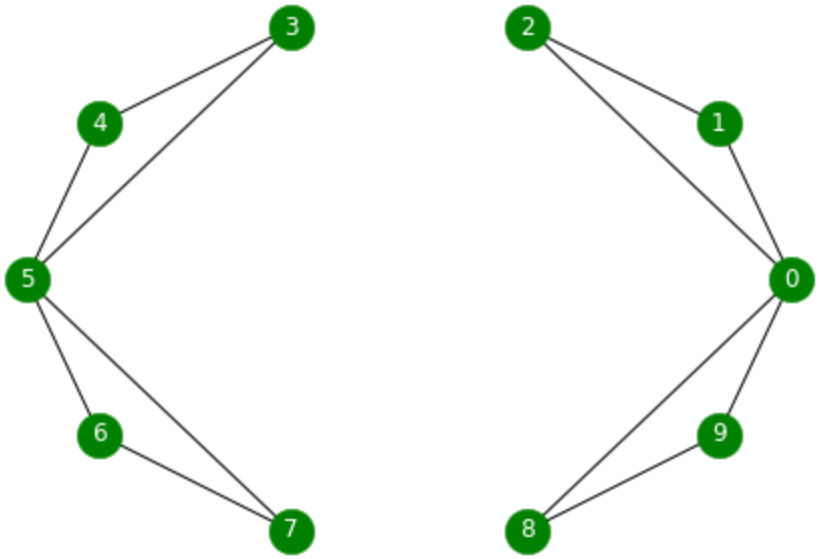
Γράφημα

- Αυτό το γράφημα έχει 10 κόμβους και 12 ακμές.
- Έχει επίσης δύο συνδεδεμένα στοιχεία {0,1,2,8,9} και {3,4,5,6,7}
- Τα συνδεδεμένα στοιχεία είναι υποψήφια για cluster, αλλά μικρότερες δομές μπορεί επίσης να είναι υποψήφιες



Adjacency Matrix

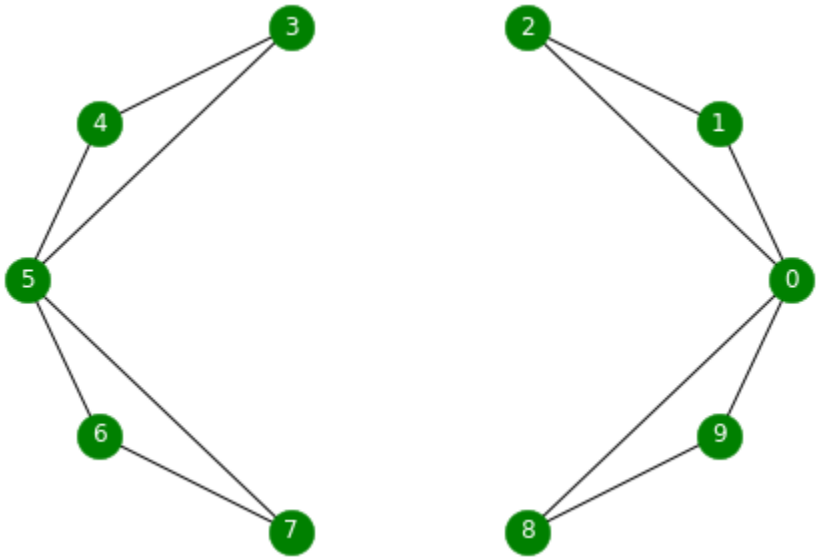
```
A = np.array([
  [0, 1, 1, 0, 0, 0, 0, 0, 1, 1],
  [1, 0, 1, 0, 0, 0, 0, 0, 0, 0],
  [1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 1, 1, 0, 0, 0, 0],
  [0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
  [0, 0, 0, 1, 1, 0, 1, 1, 0, 0],
  [0, 0, 0, 0, 0, 1, 0, 1, 0, 0],
  [0, 0, 0, 0, 0, 1, 1, 0, 0, 0],
  [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 0, 0, 0, 0, 0, 0, 1, 0]])
```



- Connected to => 1, else 0

Degree Matrix

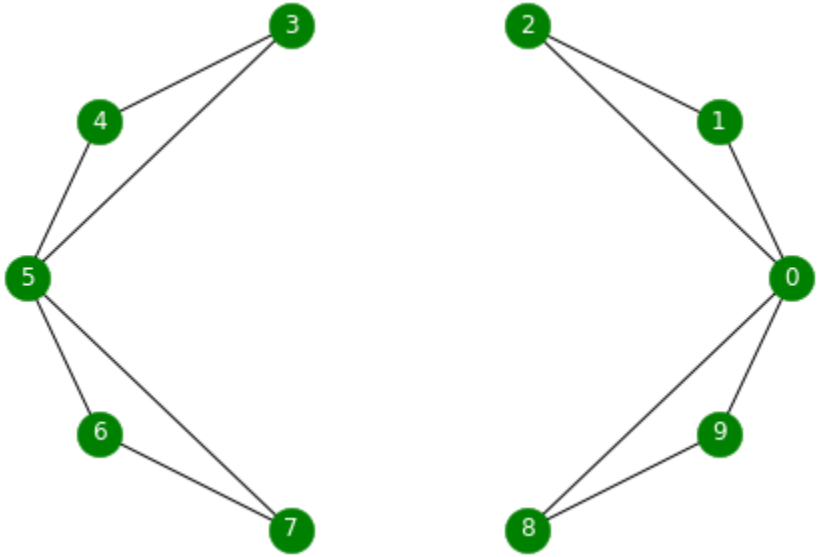
```
1   D = np.diag(A.sum(axis=1))
2   print(D)
3
4   # [[4 0 0 0 0 0 0 0 0 0]
5   #  [0 2 0 0 0 0 0 0 0 0]
6   #  [0 0 2 0 0 0 0 0 0 0]
7   #  [0 0 0 2 0 0 0 0 0 0]
8   #  [0 0 0 0 2 0 0 0 0 0]
9   #  [0 0 0 0 0 4 0 0 0 0]
10  #  [0 0 0 0 0 0 2 0 0 0]
11  #  [0 0 0 0 0 0 0 2 0 0]
12  #  [0 0 0 0 0 0 0 0 2 0]
13  #  [0 0 0 0 0 0 0 0 0 2]]
```

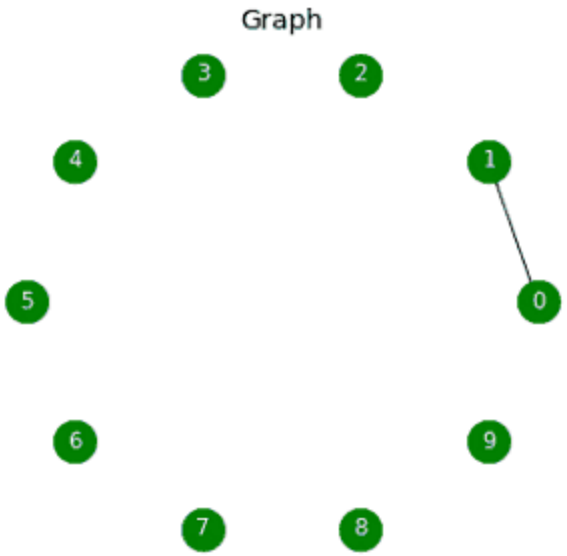


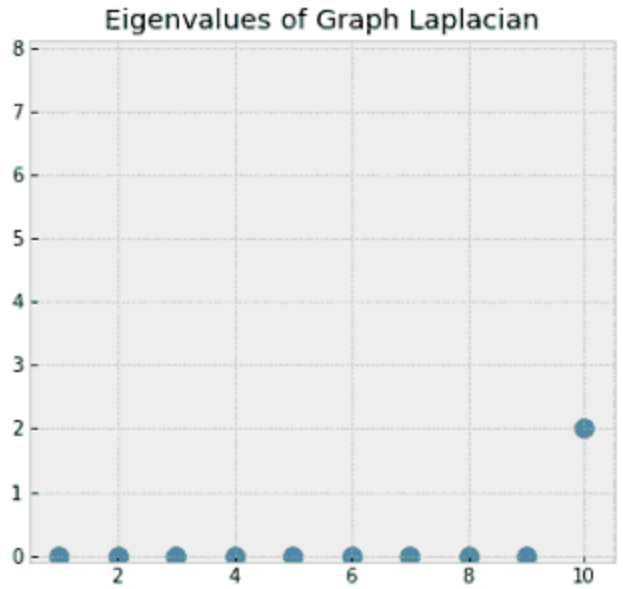
- Διαγώνιος με στοιχεία τον αριθμό ακμών από τον αντίστοιχο κόμβο

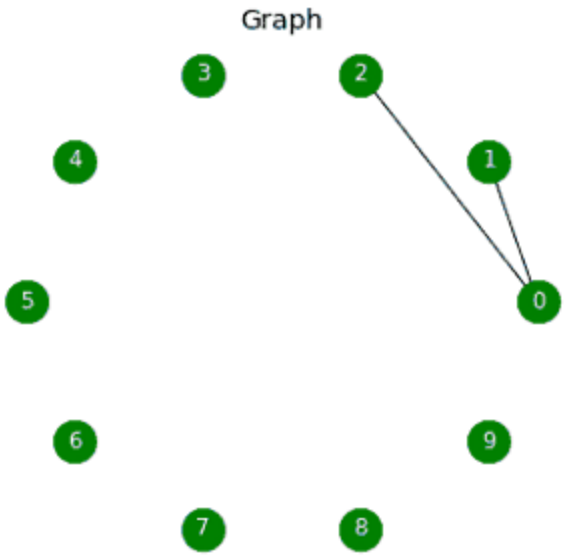
Graph Laplacian

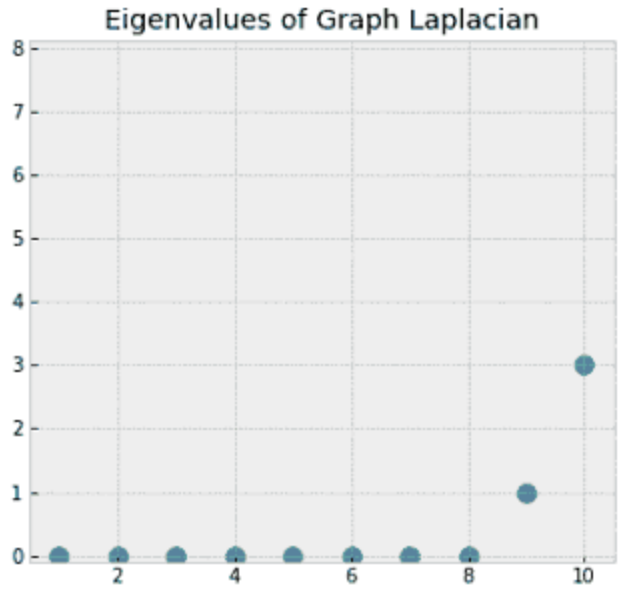
```
1   L = D-A
2   print(L)
3
4   # [[ 4 -1 -1  0  0  0  0  0 -1 -1]
5   #  [-1  2 -1  0  0  0  0  0  0  0]
6   #  [-1 -1  2  0  0  0  0  0  0  0]
7   #  [ 0  0  0  2 -1 -1  0  0  0  0]
8   #  [ 0  0  0 -1  2 -1  0  0  0  0]
9   #  [ 0  0  0 -1 -1  4 -1 -1  0  0]
10  #  [ 0  0  0  0  0 -1  2 -1  0  0]
11  #  [ 0  0  0  0  0 -1 -1  2  0  0]
12  #  [-1  0  0  0  0  0  0  0  2 -1]
13  #  [-1  0  0  0  0  0  0  0 -1  2]]
```

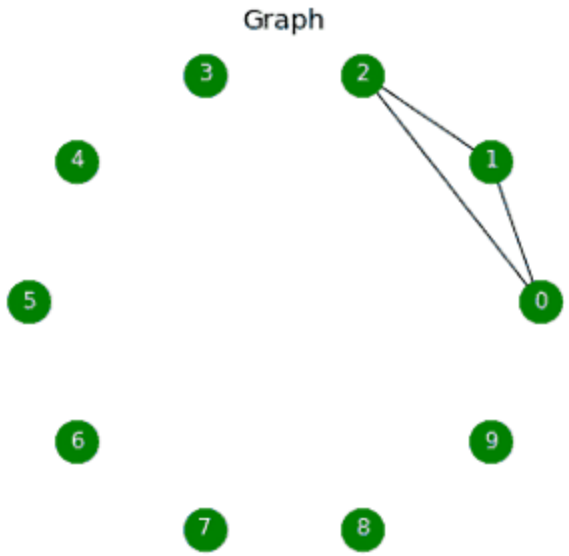


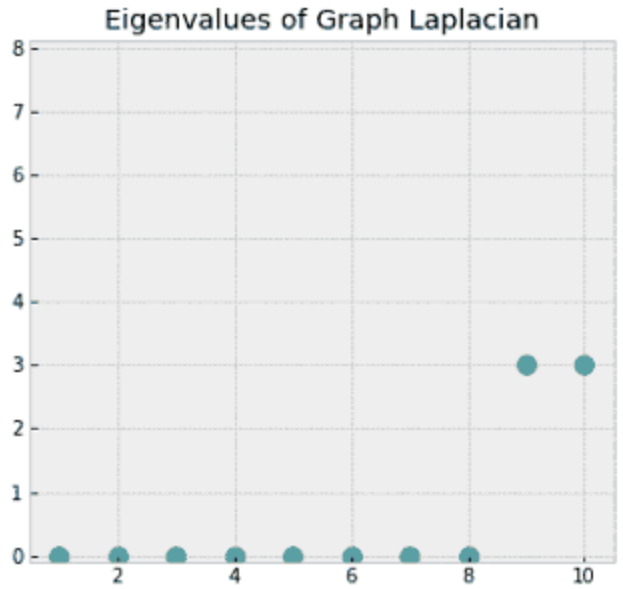

Graph
3
2
4
1
5
0
6
9
7
8

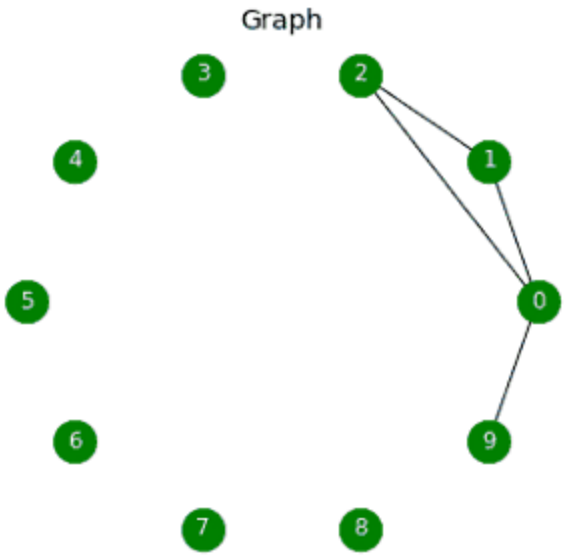

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

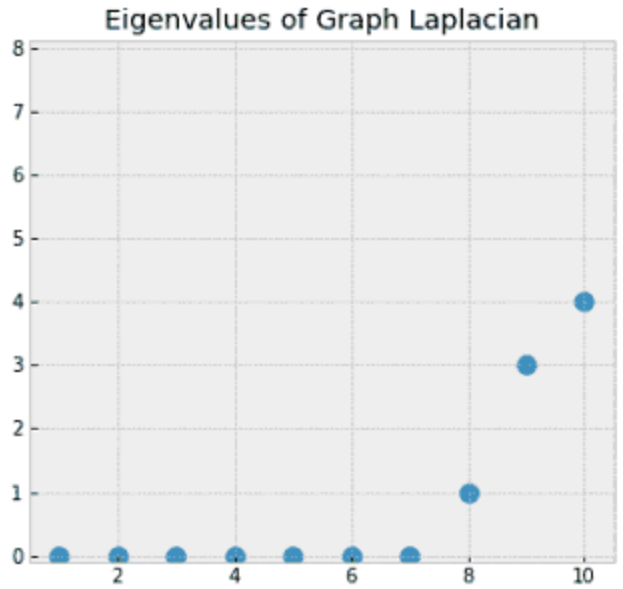

Graph
3
2
4
1
5
0
6
9
7
8

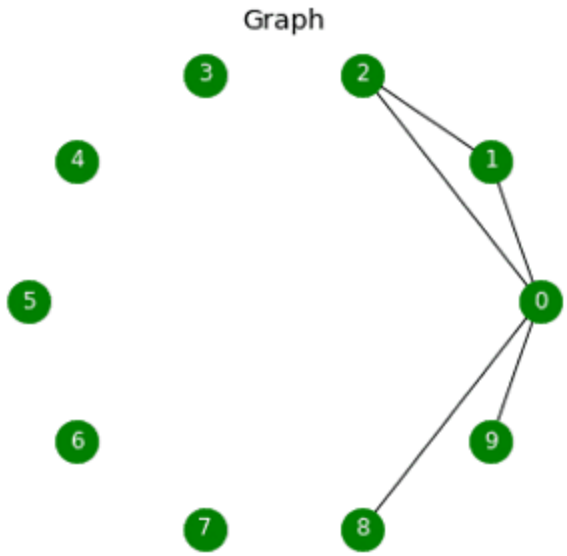

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

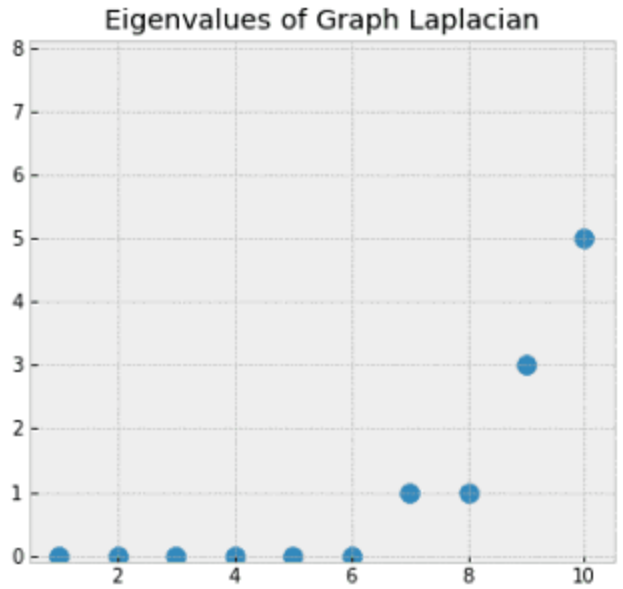

Graph
3
2
4
1
5
0
6
9
7
8

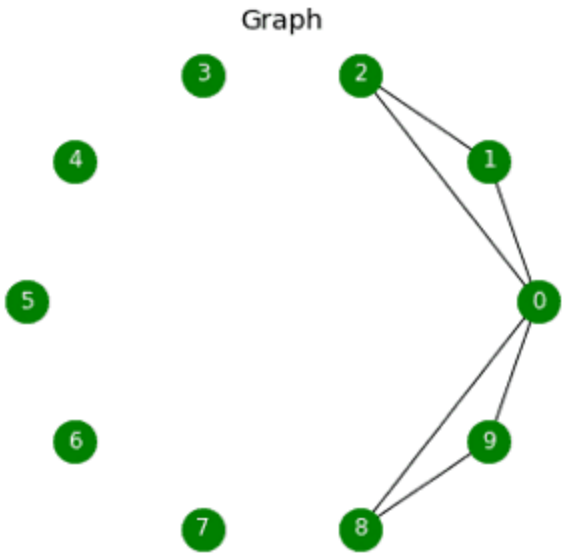

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

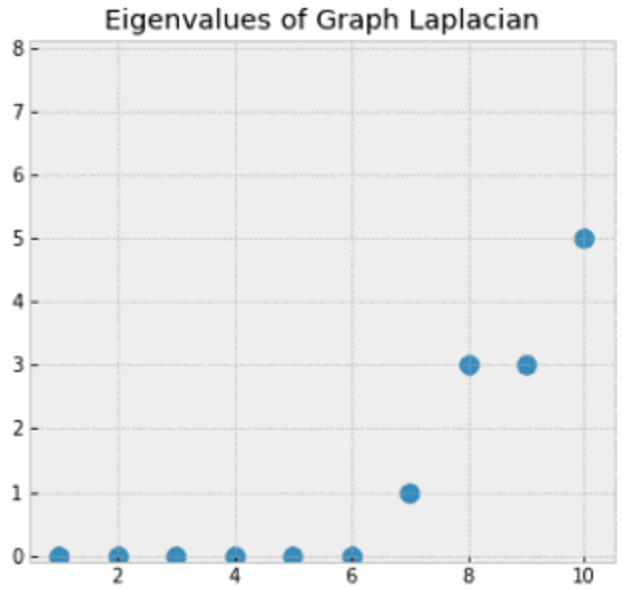

Graph
3
2
4
1
5
0
6
9
7
8

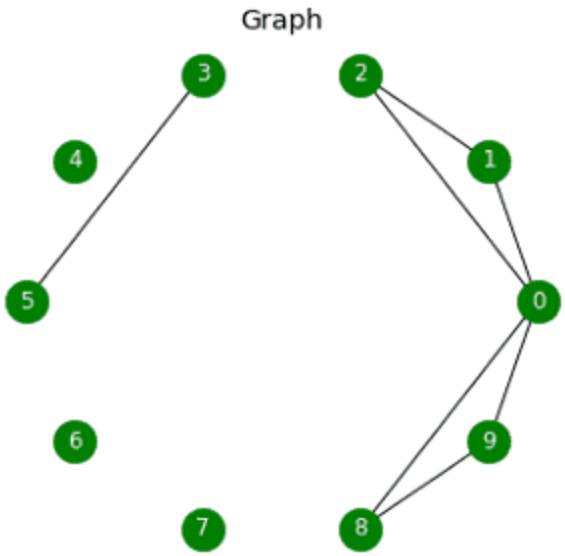

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

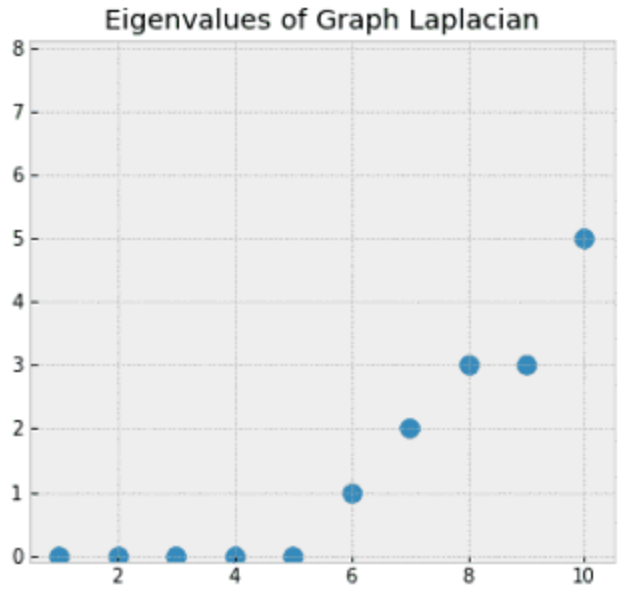

Graph
3
2
4
1
5
0
6
9
7
8

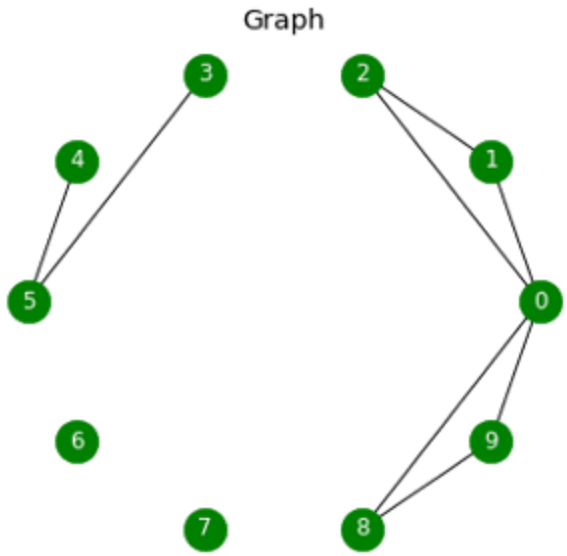

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

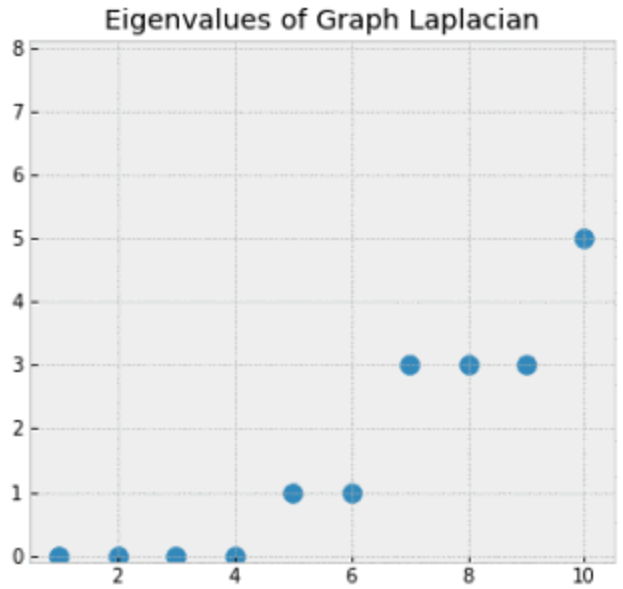

Graph
3
2
4
1
5
0
6
9
7
8

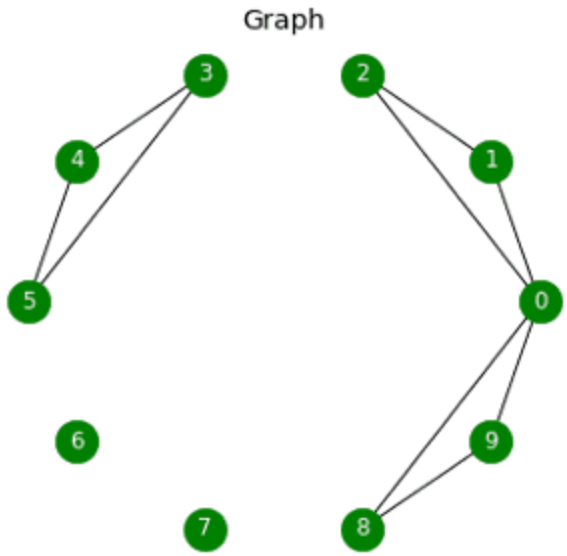

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

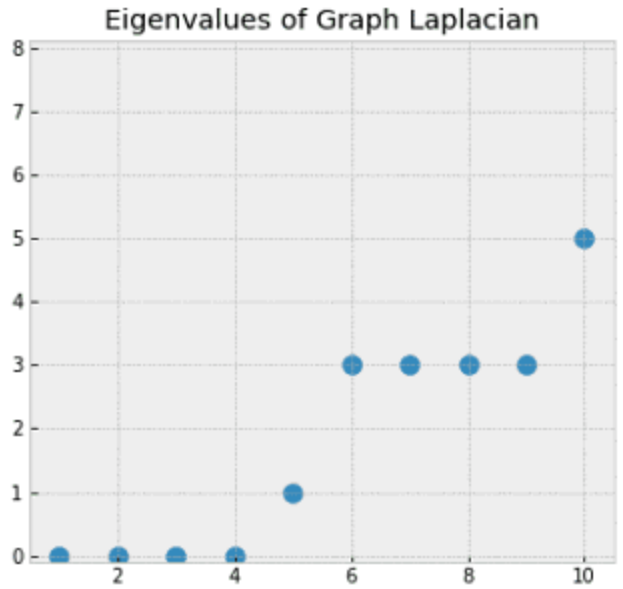

Graph
3
2
4
1
5
0
6
9
7
8

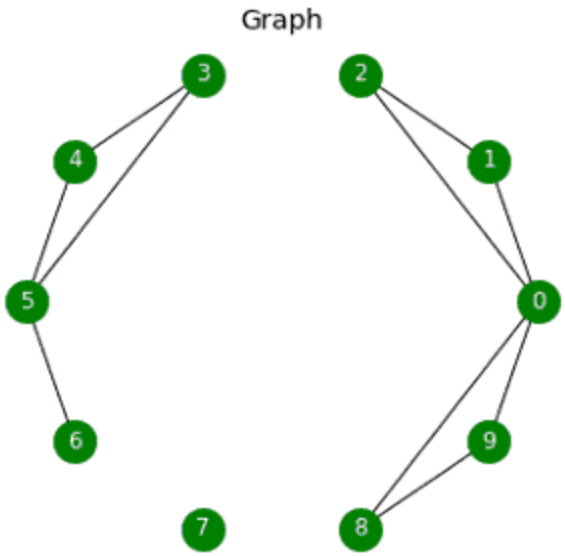

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

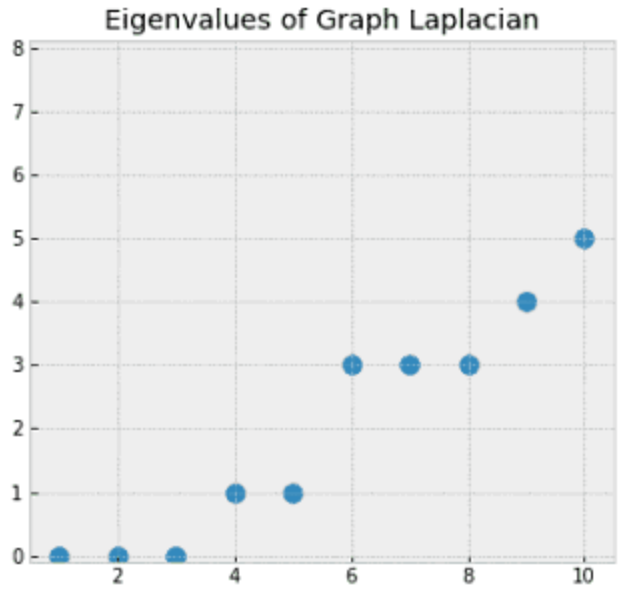

Graph
3
2
4
1
5
0
6
9
7
8

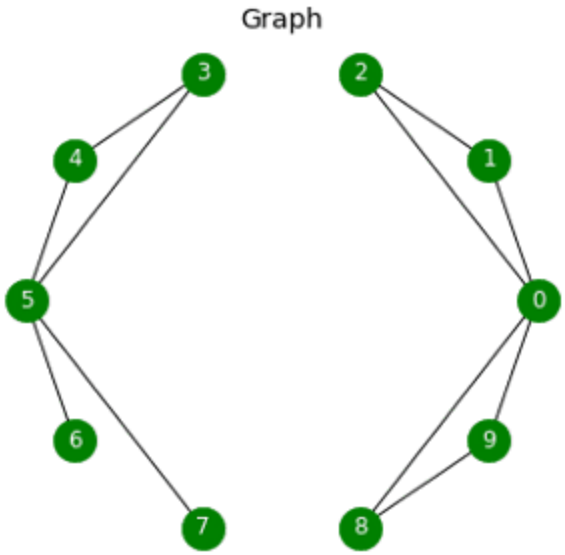

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

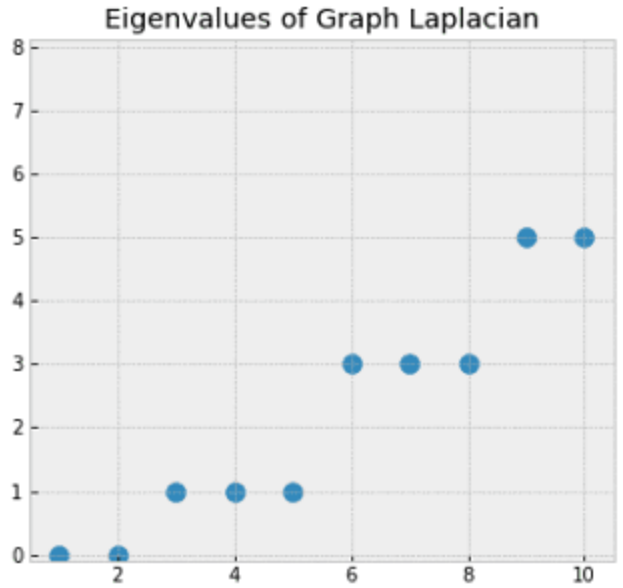

Graph
3
2
4
1
5
0
6
9
7
8

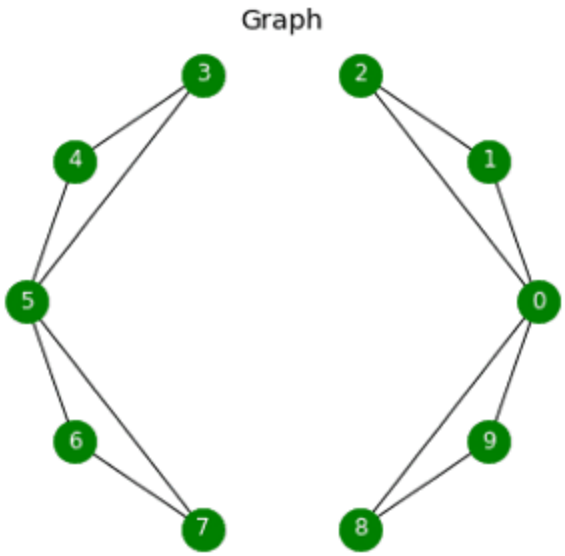

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

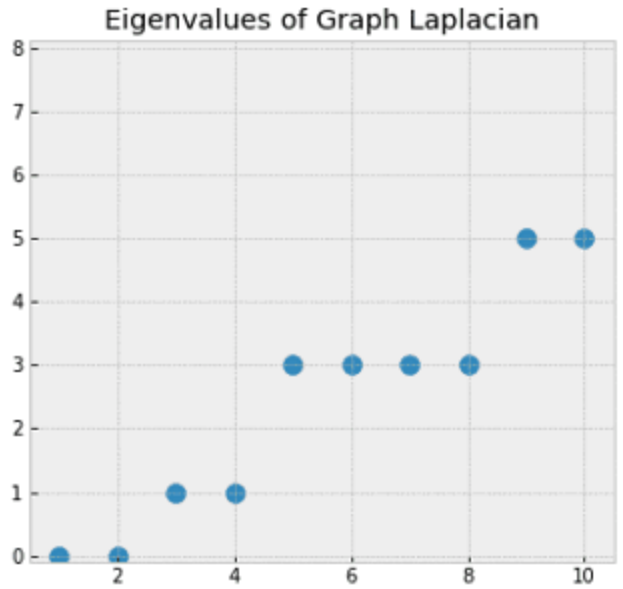

Graph
3
2
4
1
5
0
6
9
7
8

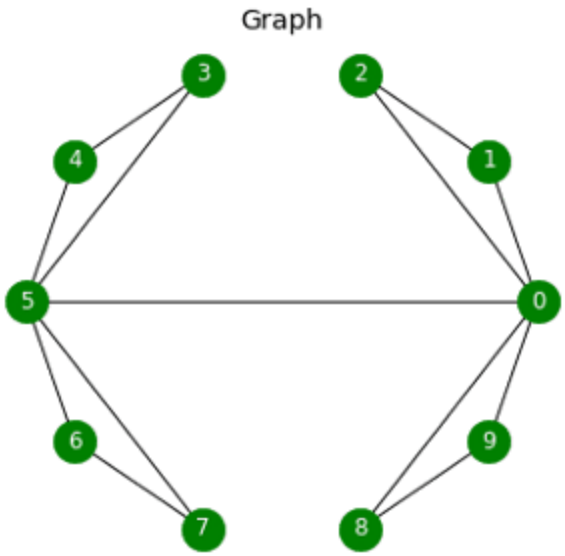

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

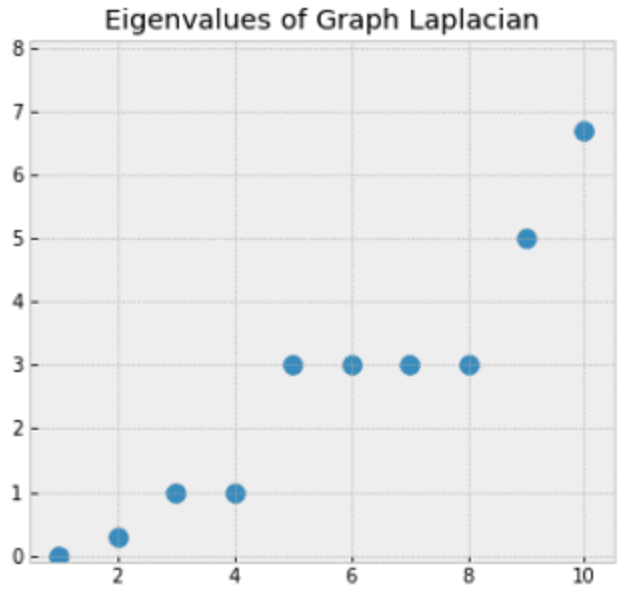

Graph
3
2
4
1
5
0
6
9
7
8

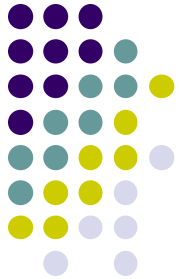

Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10


Graph
3
2
4
1
5
0
6
9
7
8


Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

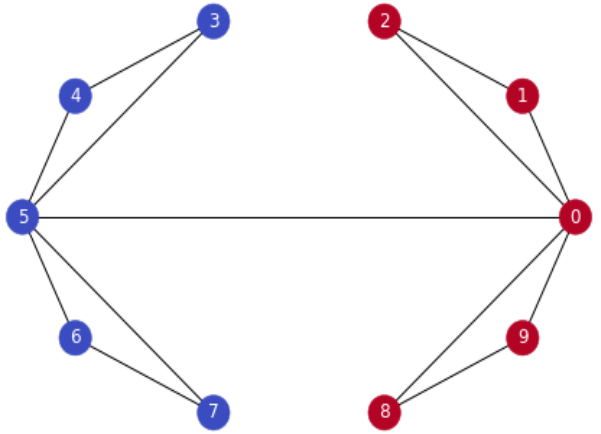

Graph
3
2
4
1
5
0
6
9
7
8


Eigenvalues of Graph Laplacian
8
7
6
5
4
3
2
1
0
2
4
6
8
10

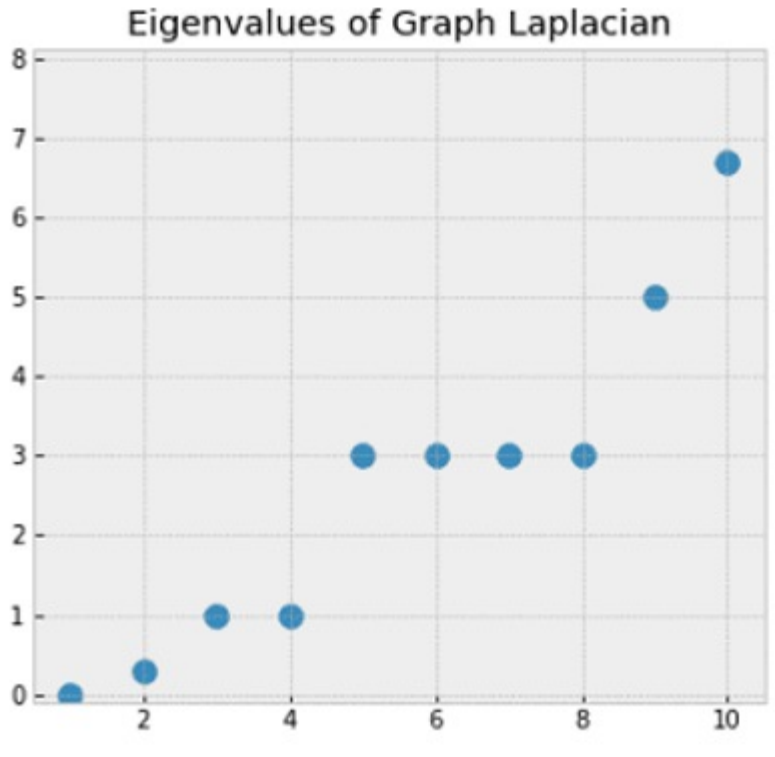


- Όταν το γράφημα αποσυνδεθεί εντελώς, και οι δέκα ιδιοτιμές είναι 0
- Καθώς προσθέτουμε ακμές, μερικές από τις ιδιοτιμές αυξάνονται
- Ο αριθμός των 0 ιδιοτιμών αντιστοιχεί στον αριθμό των συνδεδεμένων στοιχείων στο γράφημά μας
- Η πρώτη μη μηδενική ιδιοτιμή (φασματικό χάσμα) μας δίνει κάποια έννοια της πυκνότητας του γραφήματος (αν όλα τα ζεύγη των 10 κόμβων είχαν ακμή, τότε θα είχε τιμή 10)

- Η δεύτερη ιδιοτιμή ονομάζεται τιμή Fiedler και το αντίστοιχο διάνυσμα είναι το διάνυσμα Fiedler.
- Η τιμή Fiedler προσεγγίζει την ελάχιστη περικοπή γραφήματος που απαιτείται για να διαχωριστεί το γράφημα σε δύο συνδεδεμένα στοιχεία.
- Εάν το γράφημά μας ήταν ήδη δύο συνδεδεμένα στοιχεία, τότε η τιμή Fiedler θα ήταν 0.
- Κάθε τιμή στο ιδιοδιάνυσμα Fiedler μας δίνει πληροφορίες για το ποια πλευρά της «κοπής» ανήκει αυτός ο κόμβος.
- Ας χρωματίσουμε τους κόμβους με βάση το αν η καταχώρισή τους στο ιδιοδιάνυσμα Fielder είναι θετική ή όχι:



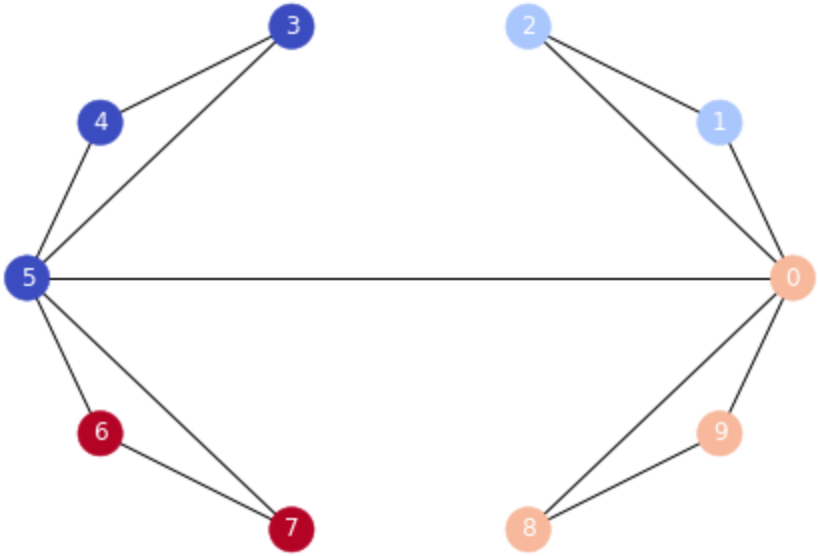
- Οι ιδιοτιμές 3,4 είναι επίσης πολύ μικρές. Αυτό μας λέει ότι είμαστε «κοντά» στο να έχουμε τέσσερις ξεχωριστά συνδεδεμένες ομάδες.
- Βρίσκουμε το πρώτο μεγάλο χάσμα μεταξύ ιδιοτιμών για να βρούμε τον αριθμό των ομάδων που εκφράζονται στα δεδομένα μας.
- Η ύπαρξη τεσσάρων ιδιοτιμών πριν από το χάσμα υποδηλώνει ότι πιθανώς υπάρχουν τέσσερις ομάδες.



- Δημιουργήστε έναν πίνακα από τις προβολές με βάση τα τρία ιδιοδιανύσματα και εκτελέστε ομαδοποίηση K-Means για να καθορίσετε τις αναθέσεις

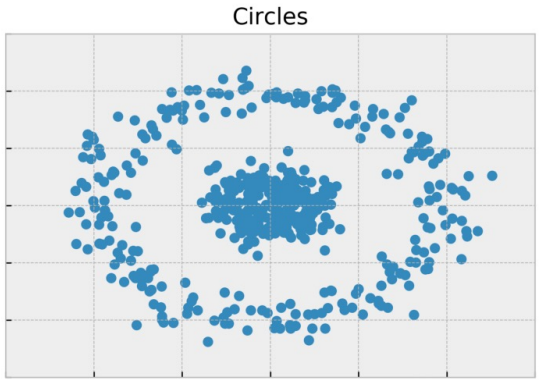
```python
import networkx as nx
import numpy as np
from scipy.linalg import eigh
from sklearn.cluster import KMeans

# Create a sample graph (you can replace this with your graph)
G = nx.karate_club_graph()

# Calculate the Laplacian matrix
L = nx.laplacian_matrix(G).astype(float)

# Compute the eigenvalues and eigenvectors
eigenvalues, eigenvectors = eigh(L.toarray())

# For k = 3, we select the eigenvectors corresponding to the 2nd, 3rd,
# since the smallest eigenvalue has an eigenvector that doesn't provide
k = 3
vectors = eigenvectors[:, 1:k+1]

# Perform k-means clustering on these vectors
kmeans = KMeans(n_clusters=k)
clusters = kmeans.fit_predict(vectors)

# Assign the cluster labels back to the nodes in the graph
for i, node in enumerate(G.nodes()):
    G.nodes[node]['cluster'] = clusters[i]
```

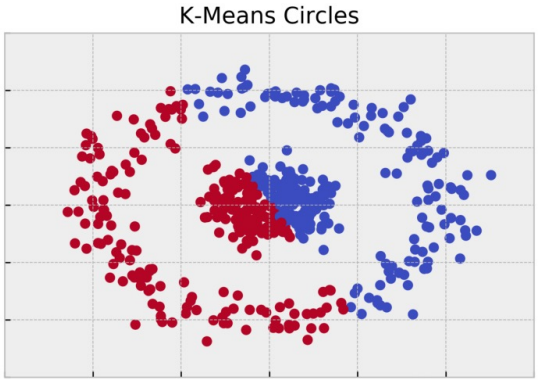
```
vecs[:,1:4]
array([[-4.47213595e-01, -5.38957608e-32, -4.05302043e-32],
       [-4.47213595e-01,  5.00000000e-01,  2.52175829e-01],
       [-4.47213595e-01,  5.00000000e-01,  2.52175829e-01],
       [ 0.00000000e+00,  0.00000000e+00, -4.31749177e-01],
       [ 0.00000000e+00,  0.00000000e+00, -4.31749177e-01],
       [ 0.00000000e+00,  0.00000000e+00,  8.06973109e-17],
       [ 0.00000000e+00,  0.00000000e+00,  4.31749177e-01],
       [ 0.00000000e+00,  0.00000000e+00,  4.31749177e-01],
       [-4.47213595e-01, -5.00000000e-01, -2.52175829e-01],
       [-4.47213595e-01, -5.00000000e-01, -2.52175829e-01]])
```

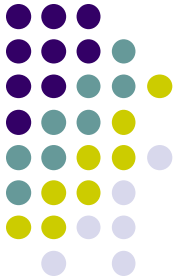


Παράδειγμα

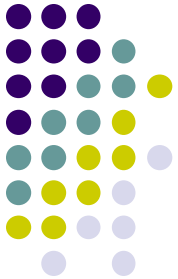
- Εφαρμογή k-means με k=2
- Ευκλείδεια απόσταση υποθέτει σφαιρική δομή για τα clusters



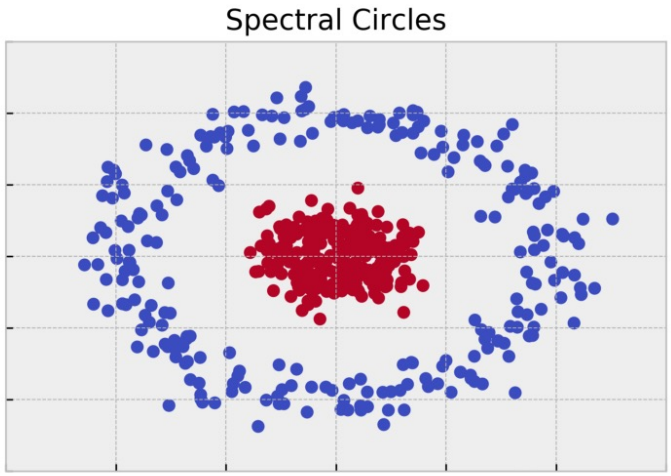




- Πώς μπορούμε να χρησιμοποιήσουμε την θεωρία γράφων;
- Φτιάχνουμε γράφο γειτόνων με βάση τον k-nn π.χ. σύνδεση με 5 πλησιέστερους γείτονες
- Επειδή έχουμε μόνο 2 clusters θα χρησιμοποιήσουμε το πρόσημο της προβολής με βάση το $2^{ο}$ ιδιοδιάνυσμα (Fiedler)



```python
from sklearn.datasets import make_circles
from sklearn.neighbors import kneighbors_graph
import numpy as np

# create the data
X, labels = make_circles(n_samples=500, noise=0.1, factor=.2)

# use the nearest neighbor graph as our adjacency matrix
A = kneighbors_graph(X, n_neighbors=5).toarray()
print(A)

# [[0. 0. 0. ... 0. 0. 0.]
#  [0. 0. 0. ... 0. 0. 0.]
#  [0. 0. 0. ... 0. 0. 0.]
#  ...
#  [0. 0. 0. ... 0. 1. 0.]
#  [0. 0. 0. ... 0. 0. 0.]
#  [0. 0. 0. ... 0. 0. 0.]]

# create the graph laplacian
D = np.diag(A.sum(axis=1))
L = D-A

# find the eigenvalues and eigenvectors
vals, vecs = np.linalg.eig(L)

# sort
vecs = vecs[:,np.argsort(vals)]
vals = vals[np.argsort(vals)]

# use Fiedler value to find best cut to separate data
clusters = vecs[:,1] > 0
```



Πηγή

William Fleshman

https://towardsdatascience.com/spectral-clustering-aba2640c0d5b