Ανάλυση Συστάδων (Cluster Analysis)

(για το μάθημα «Αναλυτική Δεδομένων & Μηχανική Μάθηση»)

Γιάννης Θεοδωρίδης

Εργαστήριο Επιστήμης Δεδομένων (Data Science Lab.)
www.datastories.org

2024.01

Πηγές

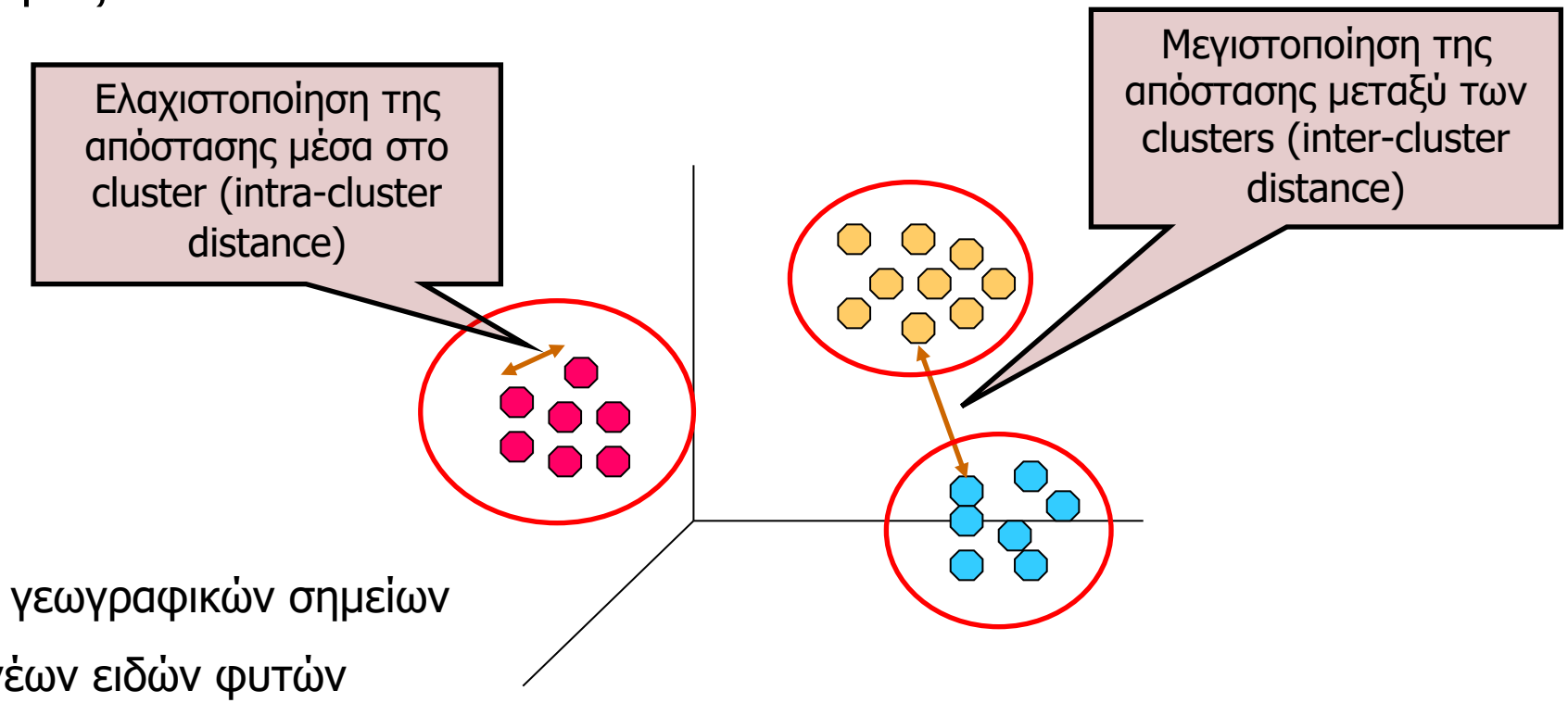
- Dunham: Data Mining – Introductory and Advanced Topics. Prentice Hall, 2003.
- Tan, Steinbach, Kumar: Introduction to Data Mining. Addison Wesley, 2006.
- Online από το διαδίκτυο κ.α.

Περιεχόμενα

- Το πρόβλημα της συσταδοποίησης
- Τεχνικές διαμέρισης (k-means)
- Τεχνικές βασισμένες στην πυκνότητα (DBSCAN, OPTICS)

Το πρόβλημα της συσταδοποίησης (1)

- **Διαμέριση (partitioning)** ενός συνόλου δεδομένων σε ομάδες (συστάδες) με βάση κάποιο κριτήριο ομοιότητας.



- Εφαρμογές
  - Ομαδοποίηση γεωγραφικών σημείων
  - Αναγνώριση νέων ειδών φυτών
  - Αναγνώριση παρόμοιων προτύπων στη χρήση του Web.

Το πρόβλημα της συσταδοποίησης (2)

- **Δεν έχει μία και μόνη λύση** – ποια είναι η βέλτιστη;

6 συστάδες (;)

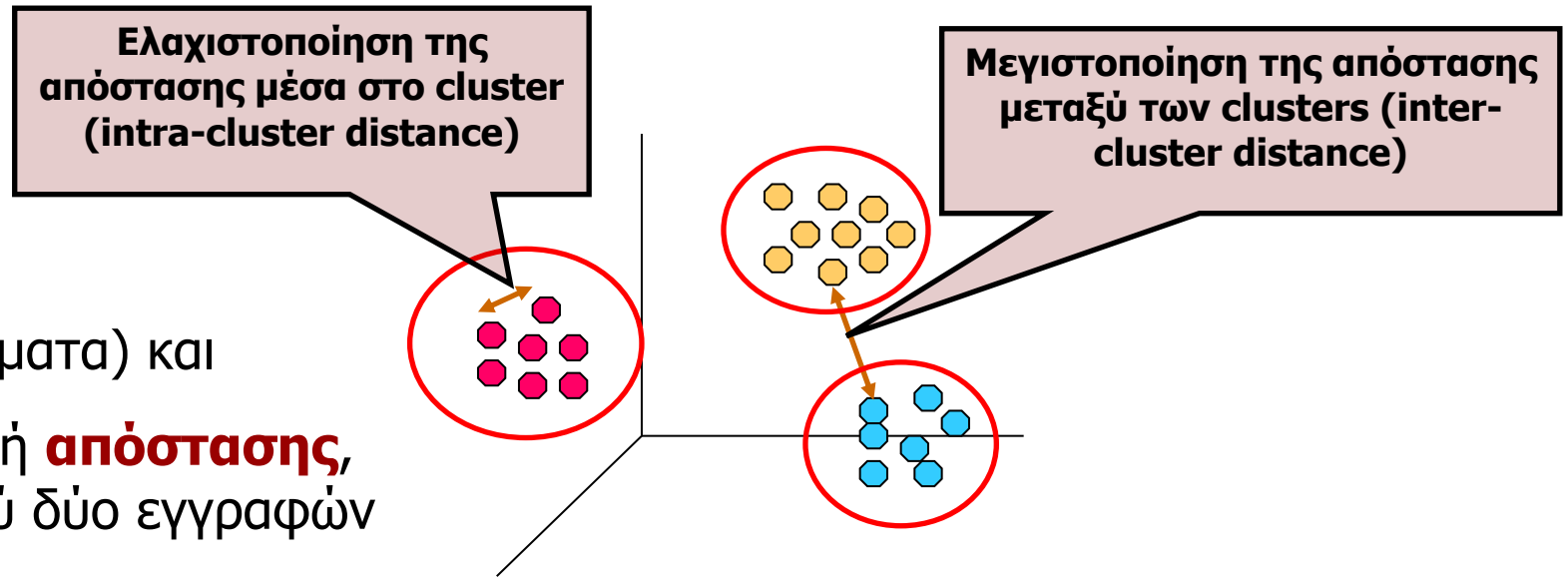
2 συστάδες (;)

4 συστάδες (;)

Το πρόβλημα της συσταδοποίησης (3)

Δοθέντων:

- ενός συνόλου δεδομένων $D=\{t_1, t_2, \ldots, t_n\}$ από n εγγραφές (στοιχεία, διανύσματα) και
- ενός **μέτρου ομοιότητας** ή **απόστασης**, $sim(t_i, t_j)$ ή $dist(t_i, t_j)$ μεταξύ δύο εγγραφών του συνόλου δεδομένων



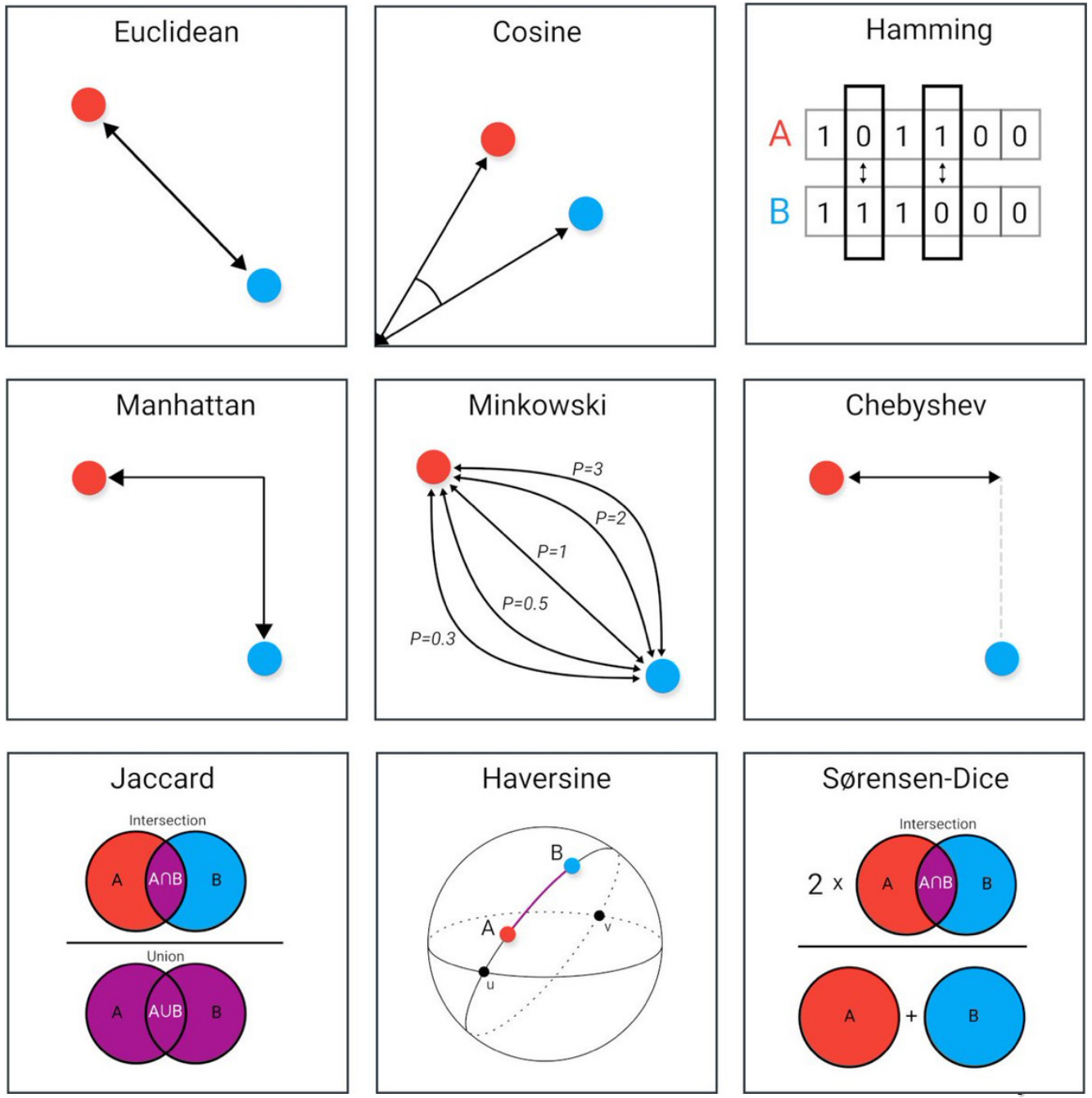
το **Πρόβλημα της Συσταδοποίησης** είναι η διαμέριση του συνόλου δεδομένων σε υποσύνολα (συστάδες), έτσι ώστε:

- να ελαχιστοποιείται η απόσταση μεταξύ των στοιχείων που ανήκουν στην ίδια συστάδα (intra-cluster distance).
- να μεγιστοποιείται η απόσταση μεταξύ των στοιχείων που ανήκουν σε διαφορετικές συστάδες (inter-cluster distance).

Μέτρα ομοιότητας / απόστασης

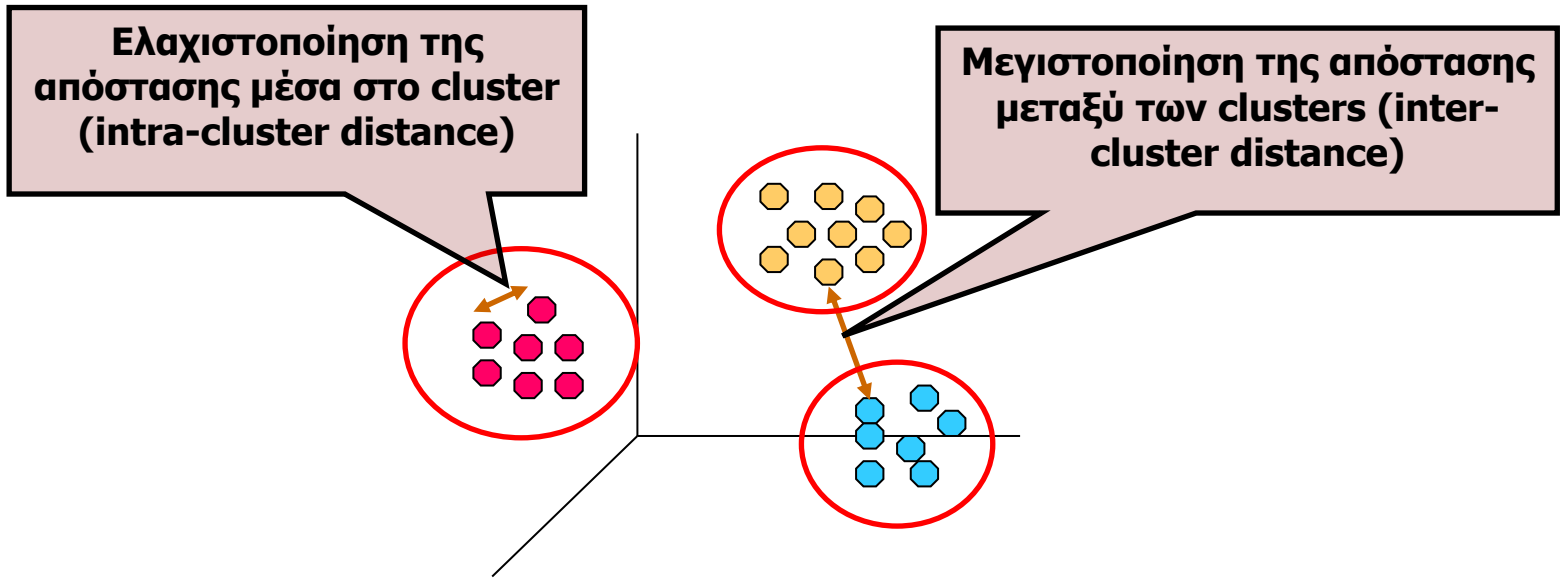
- Επιθυμητές ιδιότητες ενός μέτρου απόστασης **d** (για να ονομαστεί «**μετρική**»):
  - d(x,y) = 0 iff x=y
  - d(x,y) = d(y,x)
  - d(x,y) ≤ d(x,z) + d(z,y)

- Γιατί είναι επιθυμητό να είναι μετρική;
  - Γιατί έτσι μπορούμε να χτίσουμε ευρετήρια στον μετρικό χώρο και να επιταχύνουμε τις αναζητήσεις με βάση την απόσταση



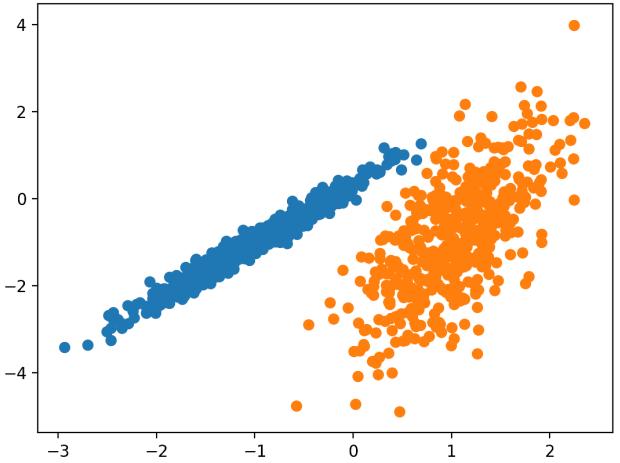
Ζητήματα στη Συσταδοποίηση

- Ποια δεδομένα θα χρησιμοποιηθούν;
  - Επιθυμούμε όλα τα δεδομένα να ενταχθούν σε συστάδες ή να μπορούν να εντοπιστούν ακραίες τιμές – «θόρυβος» (outliers / noise);
- Ποιος αλγόριθμος θα χρησιμοποιηθεί;
  - π.χ. γνωρίζουμε τον αριθμό συστάδων που στοχεύουμε;
- Πώς ερμηνεύουμε και αξιολογούμε το όποιο αποτέλεσμα;
  - π.χ. ποιο είναι το καλύτερο αποτέλεσμα; αυτό με τα 2 ή με τα 3 clusters;

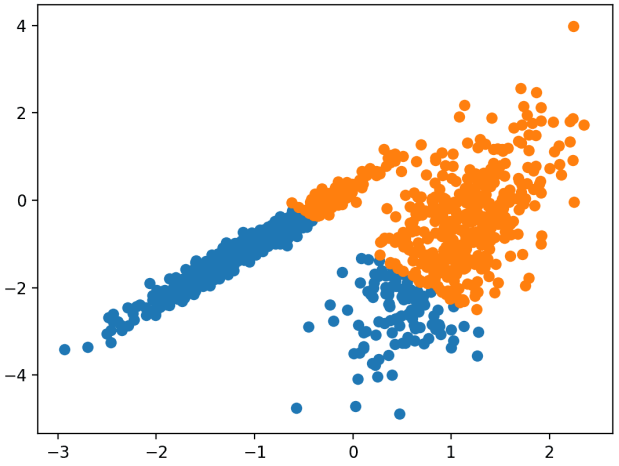


Ποιος αλγόριθμος (1);

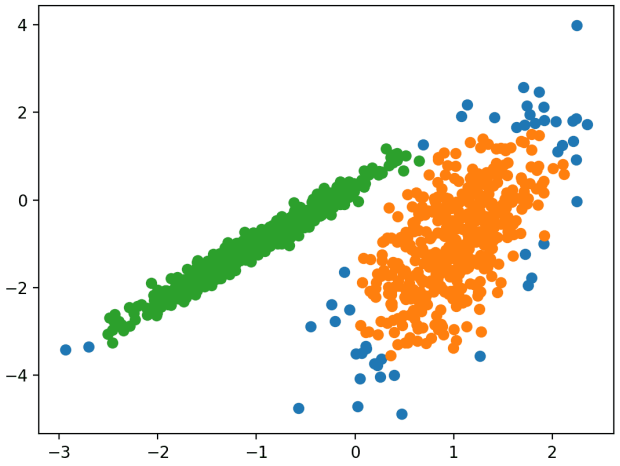
Synthetic dataset (2 clusters)



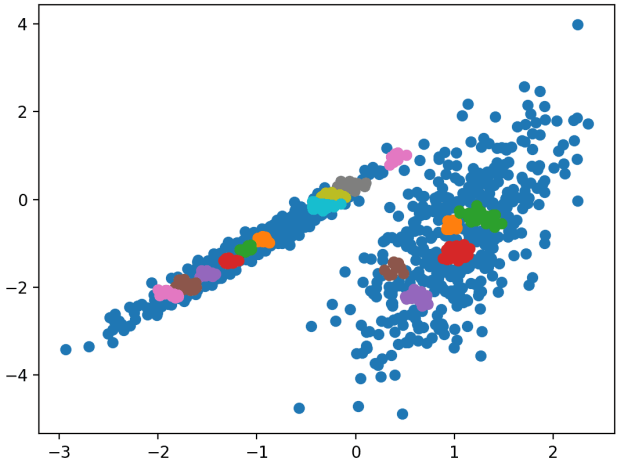
k-Means



DBSCAN

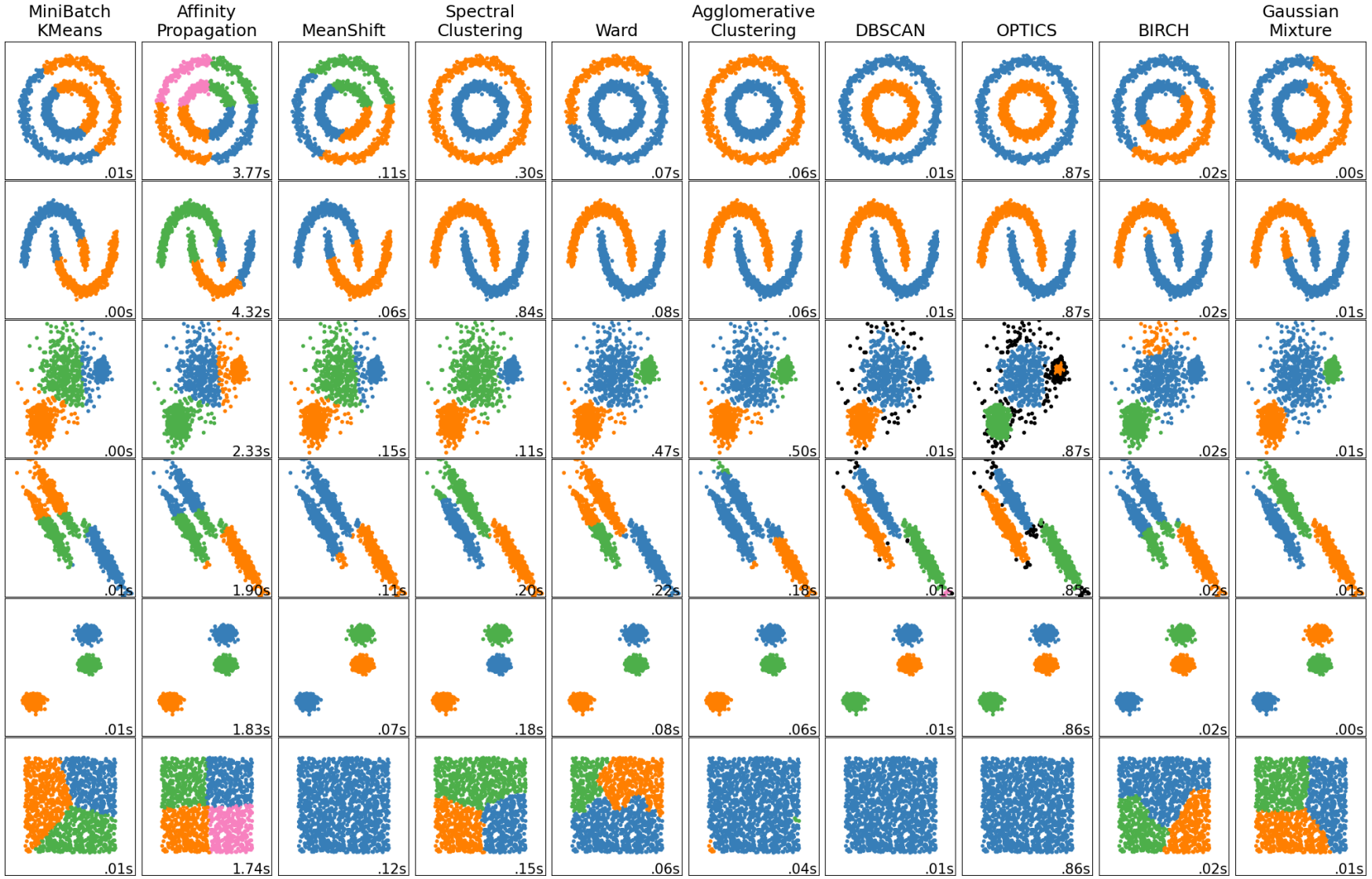


OPTICS



πηγή: https://machinelearningmastery.com/clustering-algorithms-with-python/

Ποιος αλγόριθμος (2);



πηγή: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Συσταδοποίηση με Διαμέριση

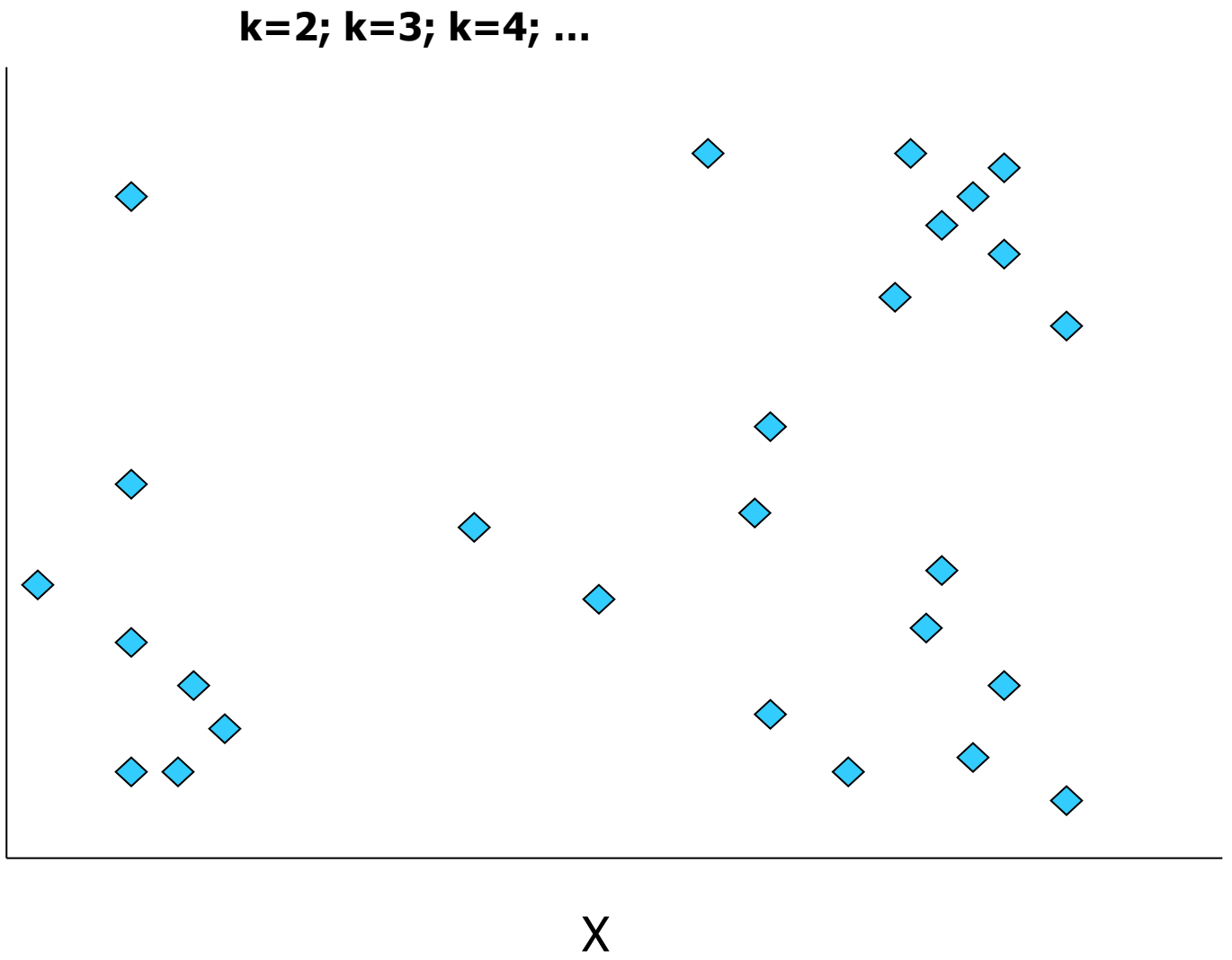
- Ζήτημα: ο χρήστης απαιτείται να εισάγει τον επιθυμητό αριθμό των συστάδων, k. Όμως …
  - Οι πιθανοί συνδυασμοί n στοιχείων σε k συστάδες είναι ένας πολύ μεγάλος αριθμός
  - Αναγκαστικά, η αναζήτηση γίνεται σε ένα μικρό υποσύνολο των πιθανών λύσεων

- Η πιο δημοφιλής τεχνική: **K-Μέσων** (K-means)
  - και πολλές άλλες τεχνικές βασισμένες σε γενετικούς αλγορίθμους, νευρωνικά δίκτυα κ.α.

Συσταδοποίηση K-Means

- Βασική ιδέα:
  - Τυχαία επιλέγεται το αρχικό σύνολο των μέσων(1) των συστάδων
  - Επαναληπτικά, τα στοιχεία μετακινούνται μεταξύ των συστάδων μέχρι να επιτύχουμε ισορροπία

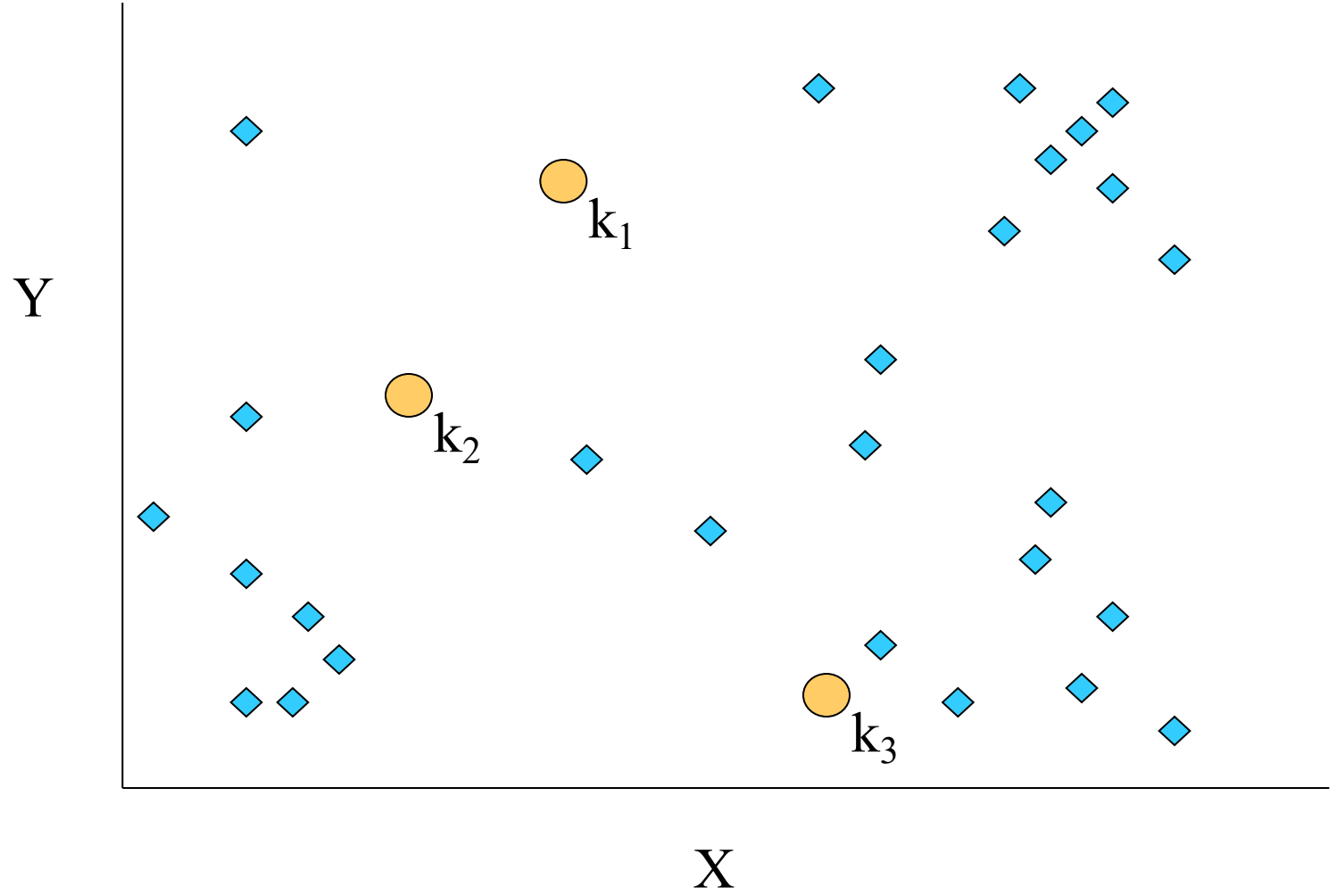
(1) Δεδομένης μίας συστάδας K = {t_1, t_2, …, t_m}, ο **μέσος ή κέντρο βάρους** (centroid) **της συστάδας** είναι m = (1/m)(t_1 + … + t_m)



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

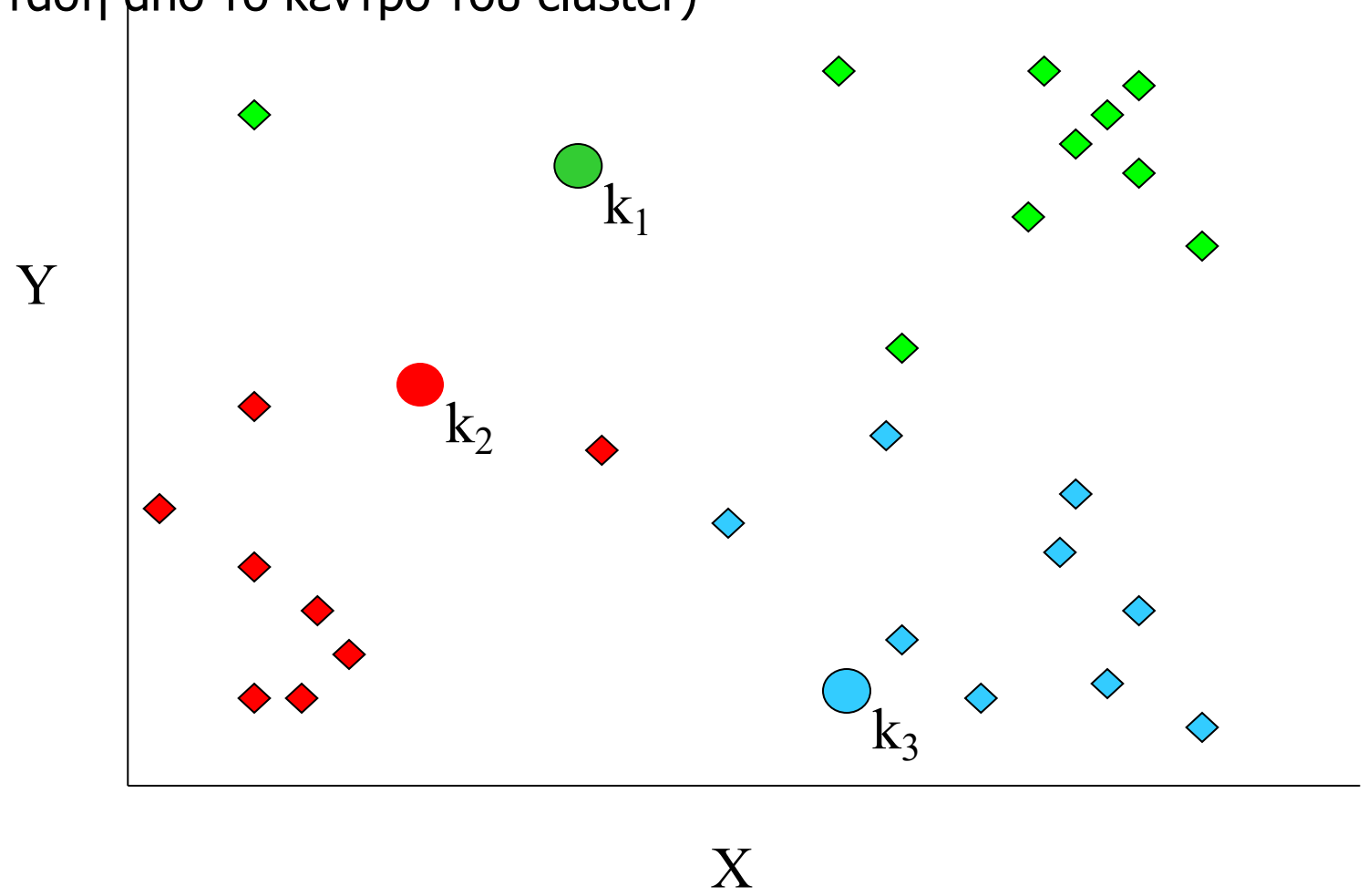
Παράδειγμα K-means (k=3)

- Τυχαία επιλογή τριών αρχικών κέντρων



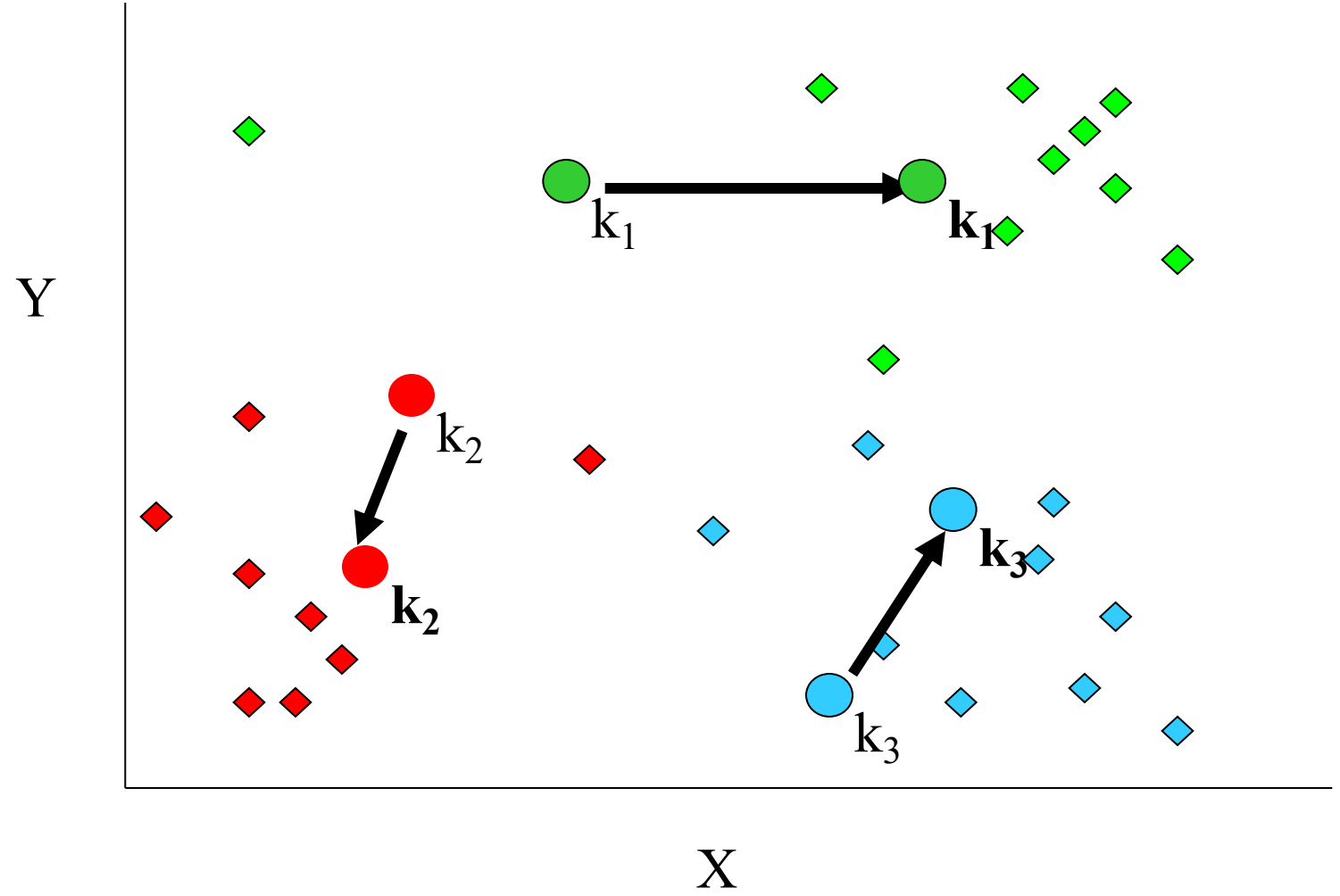
Παράδειγμα K-means, 1η επανάληψη

- Για κάθε στοιχείο, εκχώρηση στο πλησιέστερο cluster (με βάση την απόσταση από το κέντρο του cluster)



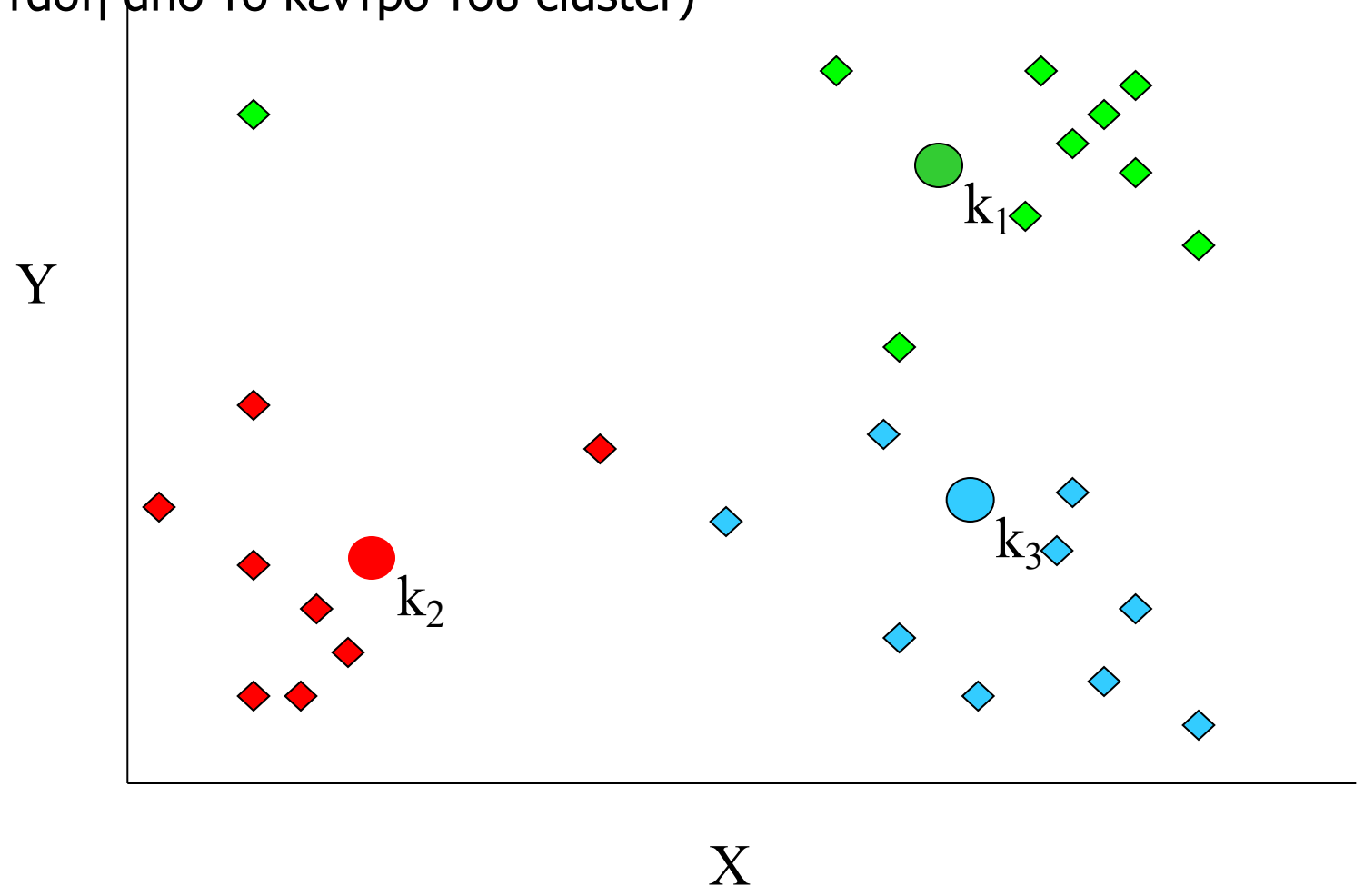
Παράδειγμα K-means, 1η επανάληψη

- Για κάθε cluster, επανυπολογισμός του νέου κέντρου βάρους

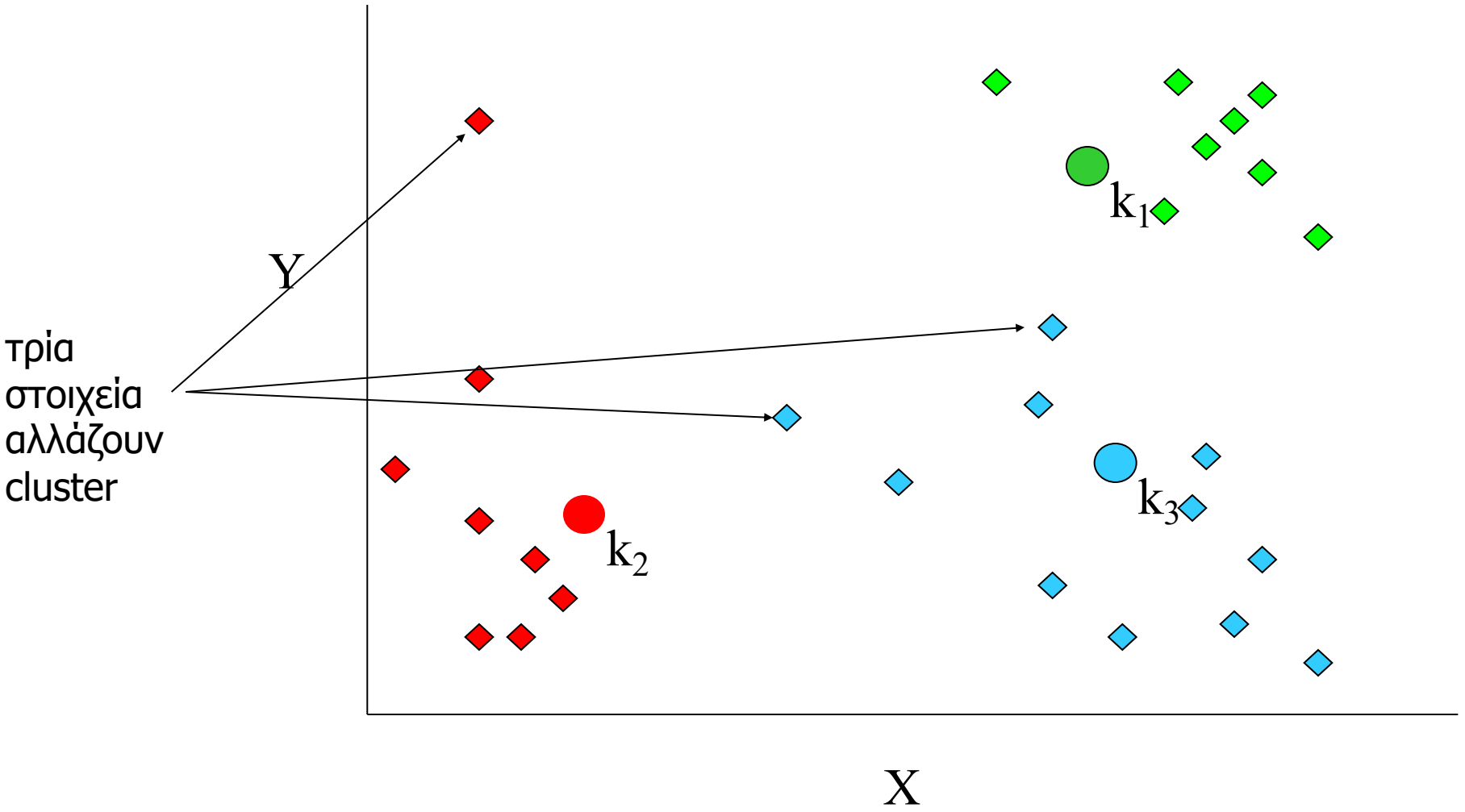


Παράδειγμα K-means, 2η επανάληψη

- Για κάθε στοιχείο, εκχώρηση στο πλησιέστερο cluster (με βάση την απόσταση από το κέντρο του cluster)

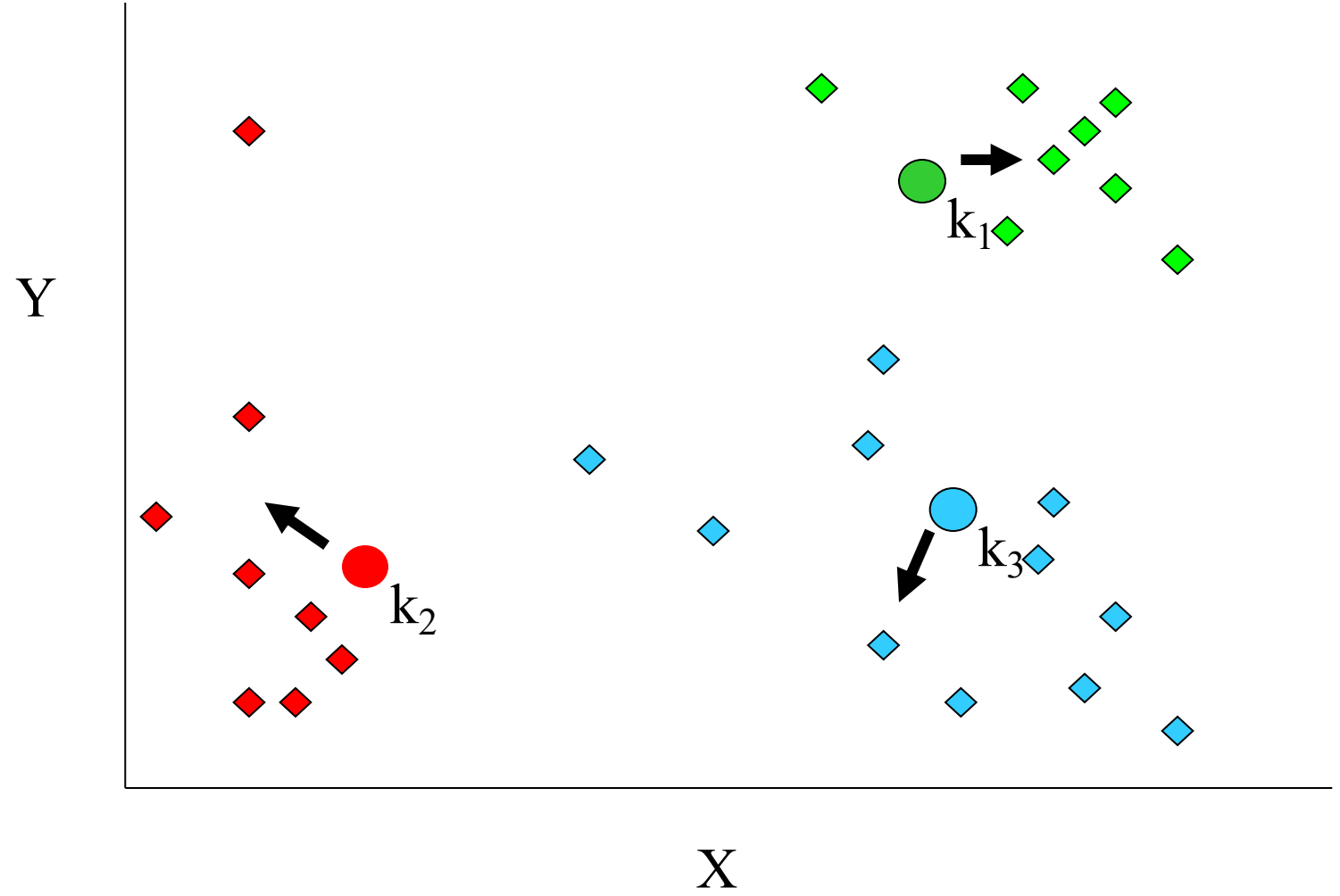


Παράδειγμα K-means, 2η επανάληψη



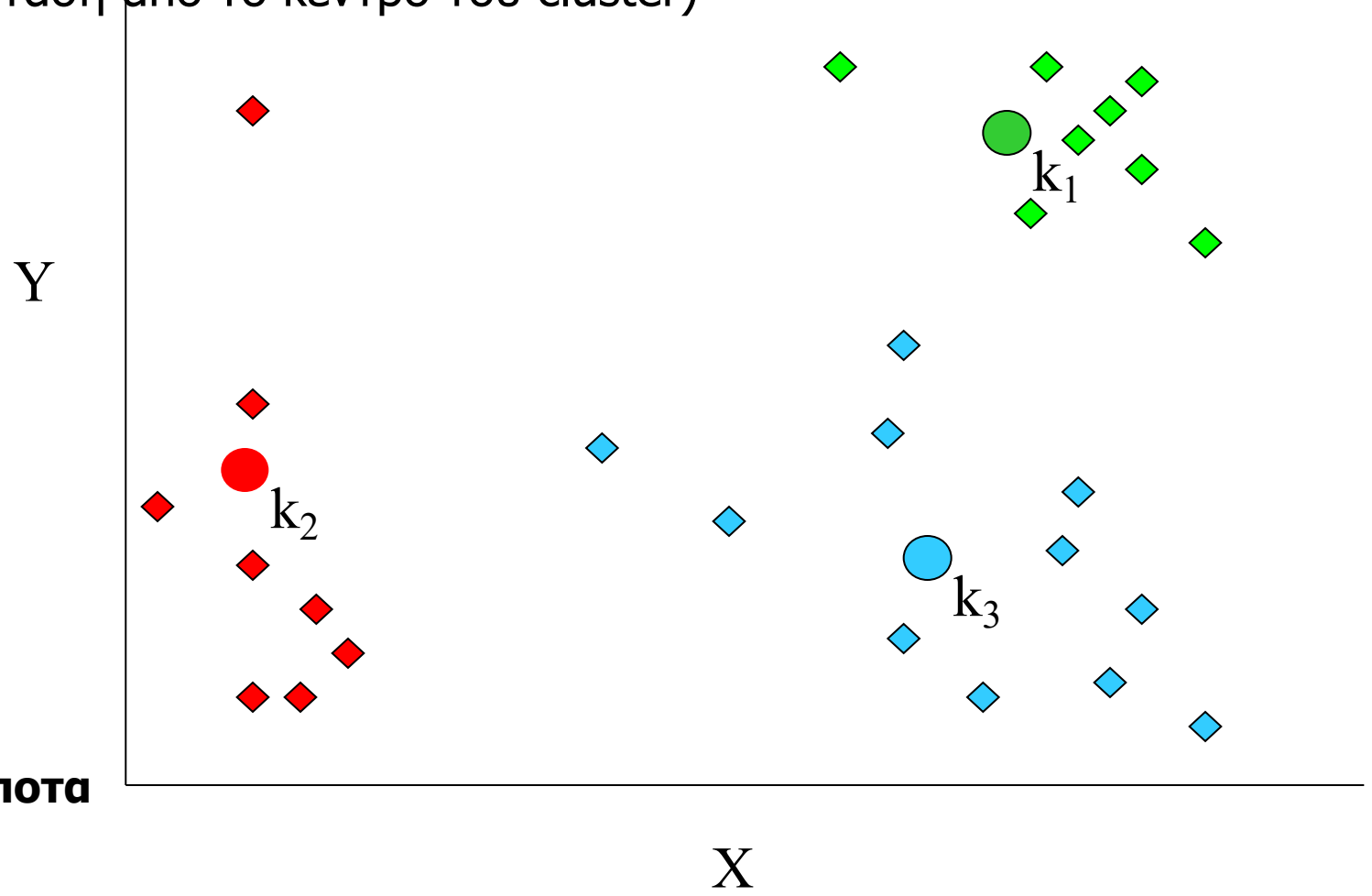
Παράδειγμα K-means, 2η επανάληψη

- Για κάθε cluster, επανυπολογισμός του νέου κέντρου βάρους



Παράδειγμα K-means, 3η επανάληψη

- Για κάθε στοιχείο, εκχώρηση στο πλησιέστερο cluster (με βάση την απόσταση από το κέντρο του cluster)



Δεν αλλάζει τίποτα
Άρα, τέλος !

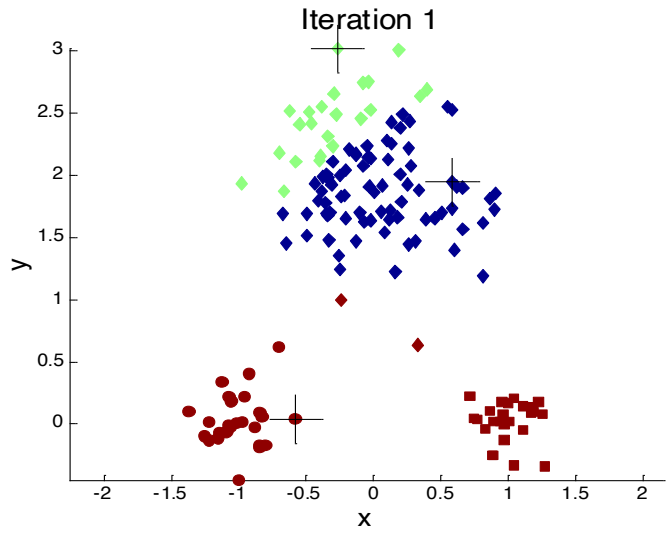
Αλγόριθμος K-Means

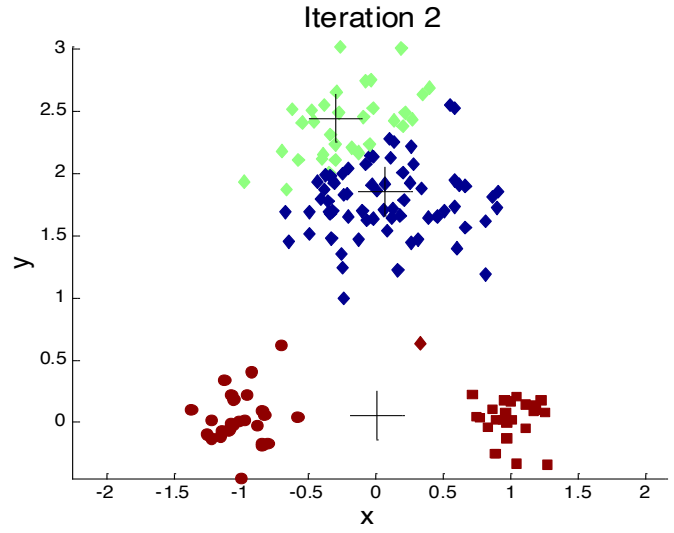
Algorithm 1 Basic K-means Algorithm.

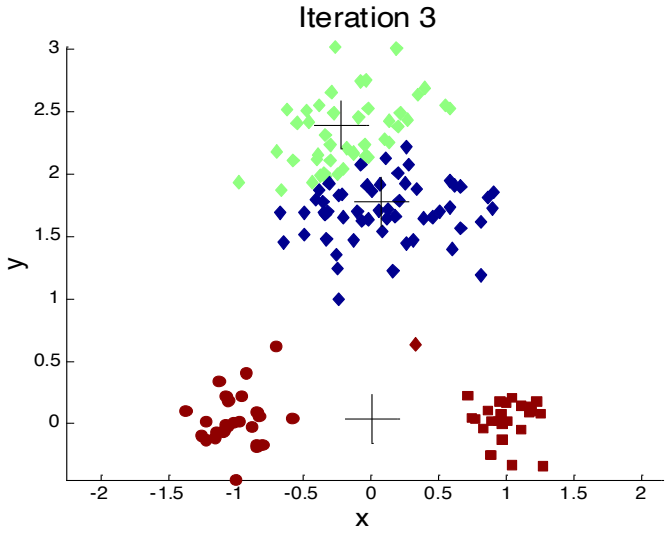
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

- Πολυπλοκότητα (βάσει του πλήθους n των στοιχείων): O(n)
- Υπέρ και κατά:
  - Ταχύς αλγόριθμος (γραμμική πολυπλοκότητα, εκτός εάν…)
  - Το πλήθος k των συστάδων πρέπει να δοθεί ως είσοδος (άρα, ποιο είναι το κατάλληλο k;)
  - Το αποτέλεσμα επηρεάζεται από την επιλογή των αρχικών μέσων
  - Οδηγεί σε «φτωχά» αποτελέσματα όταν οι συστάδες δεν έχουν «σφαιρικό» σχήμα ή τα δεδομένα περιέχουν θόρυβο (δεν υπάρχει αυτή η έννοια στον αλγόριθμο)

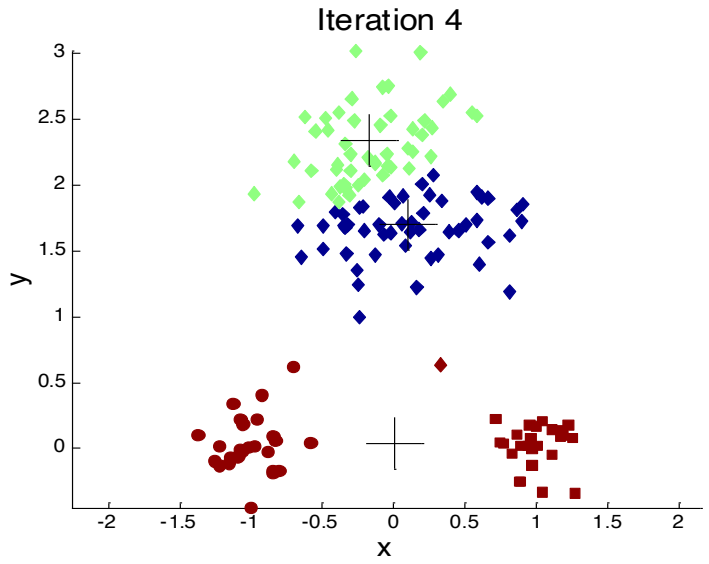
Η επίδραση της αρχικοποίησης στο αποτέλεσμα

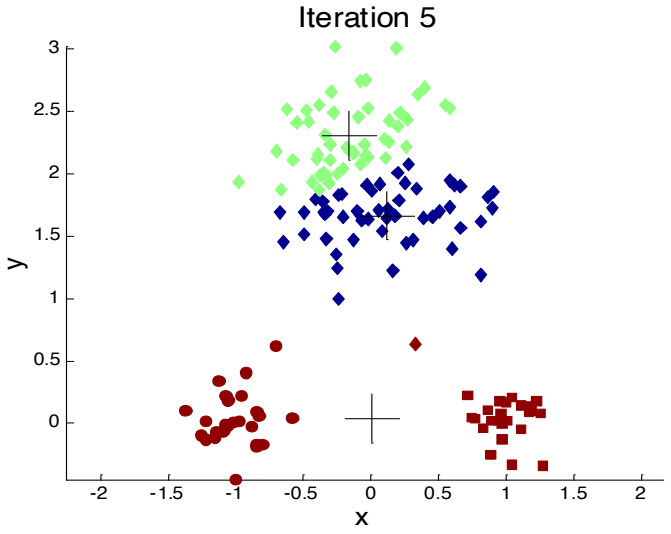






Παράδειγμα με k=3



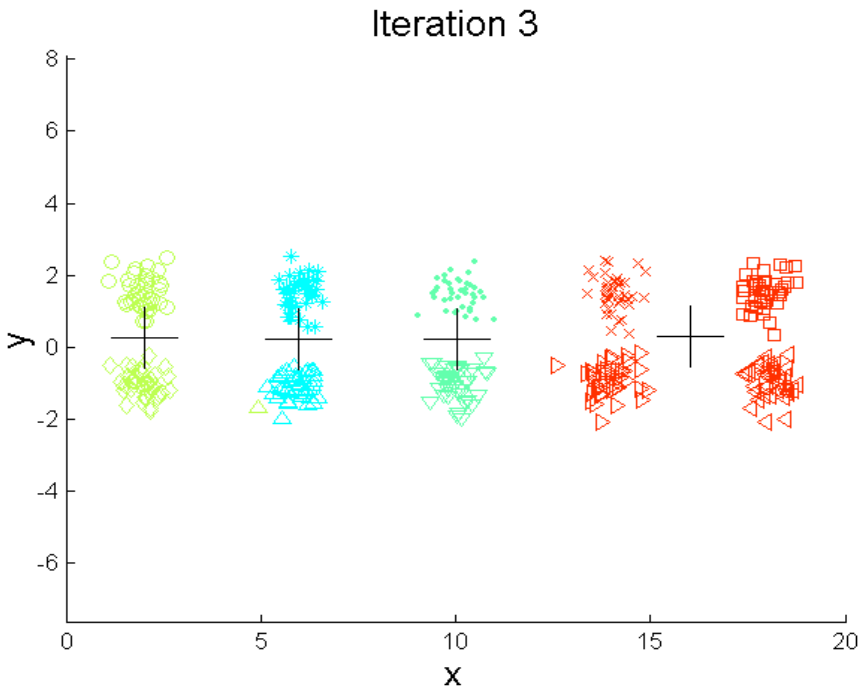


… και πώς αντιμετωπίζεται

- Πολλαπλές εκτελέσεις (με διαφορετική αρχικοποίηση)
  - Σίγουρα βοηθάει αλλά κοστίζει!
- Επιλογή αρχικών μέσων με δειγματοληψία
- **Bisecting** (διχοτομικός) **K-means**
  - Διχοτομεί κάθε φορά μια από τις υπάρχουσες συστάδες με χρήση K-means
  - Δεν παρουσιάζει τόση ευαισθησία στην αρχικοποίηση

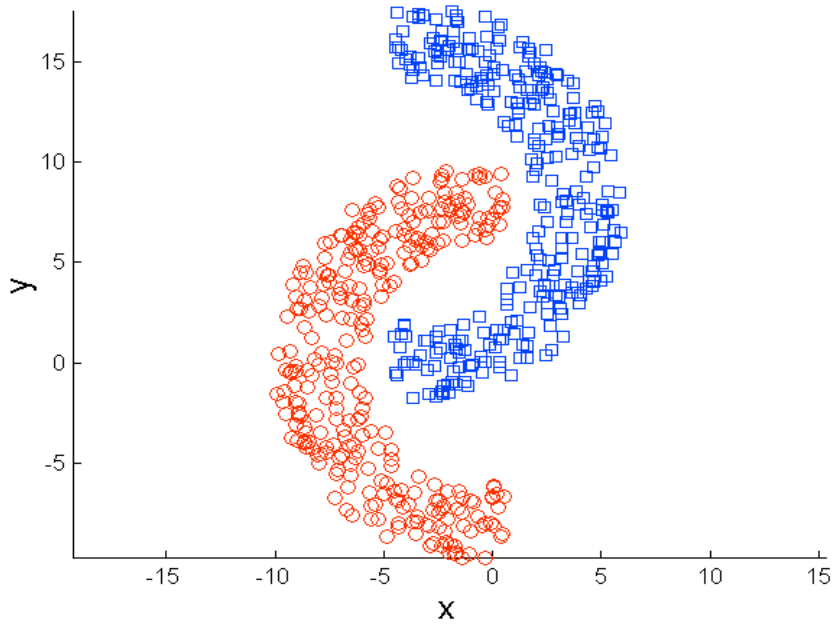
Bisecting K-means

- Παραλλαγή του K-means που παράγει διαμεριστική συσταδοποίηση με ιεραρχικό τρόπο
- Παράδειγμα: (τελικό) k=4

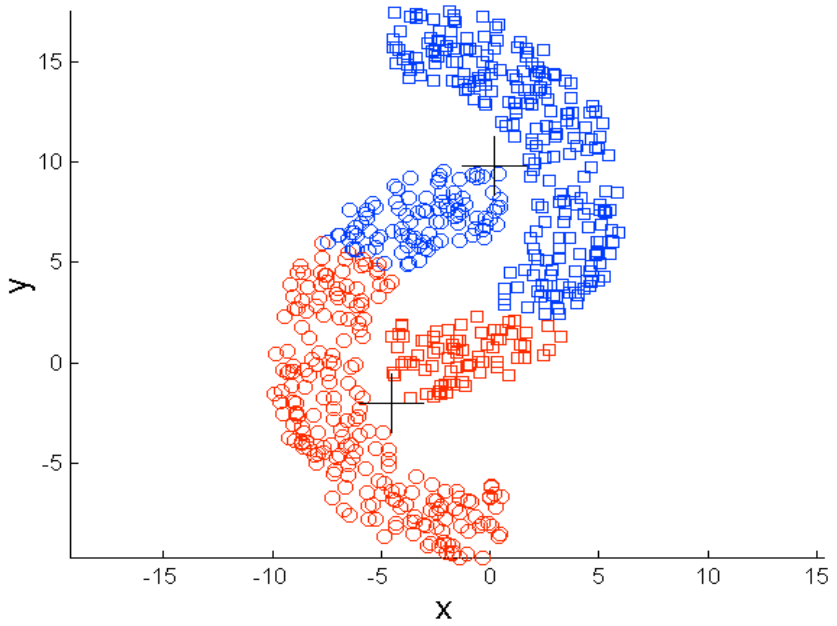


1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

Η επίπτωση του «περίεργου» σχήματος

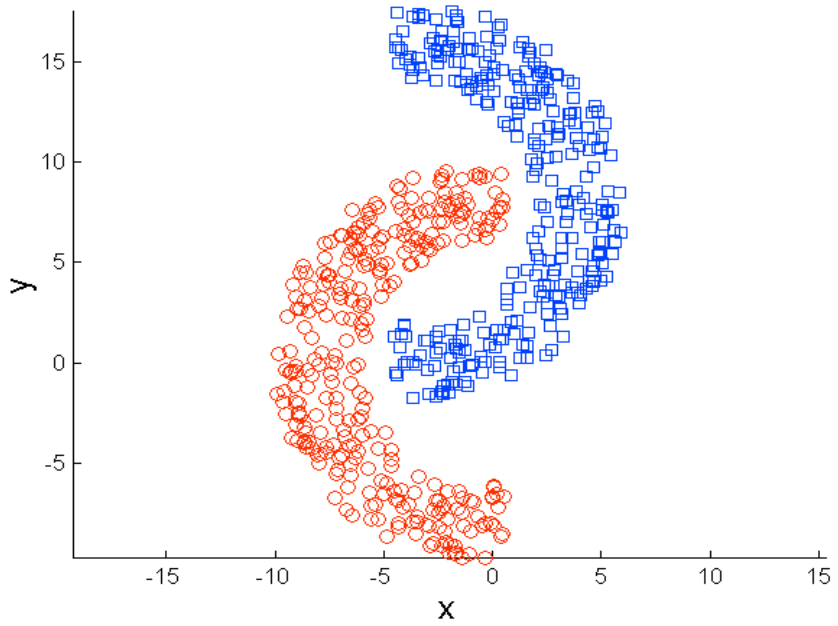


Αρχικά σημεία

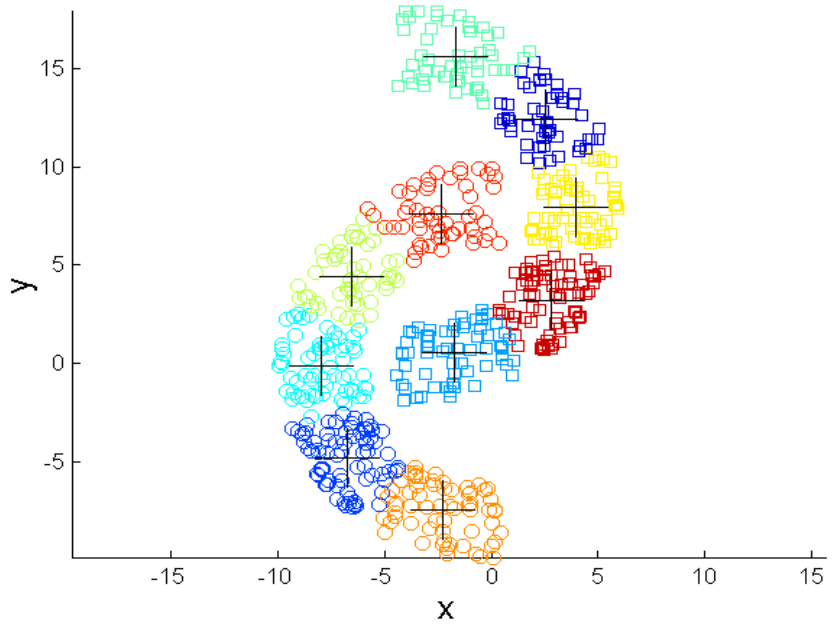


K-means (2 Clusters)

...και πώς αντιμετωπίζεται

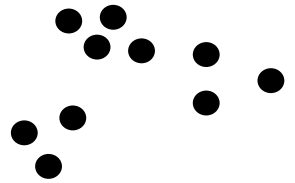


Αρχικά σημεία



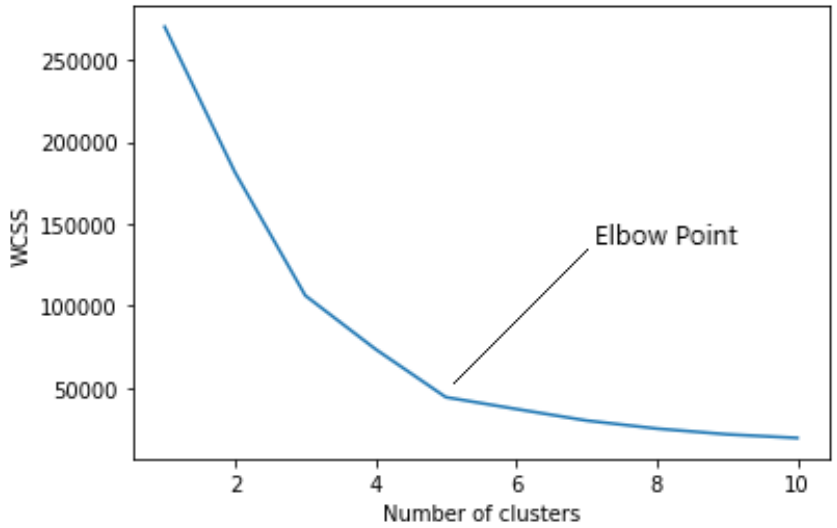
K-means (10 Clusters)

Το ζήτημα της εύρεσης του «κατάλληλου k»



- Δύσκολο πρόβλημα, χωρίς βέλτιστη λύση
- Μια ευριστική μέθοδος: το κριτήριο του «αγκώνα» (elbow criterion)
  - Για διαφορετικά k, υπολογίζουμε το μέτρο ποιότητας WCSS (Within-Cluster Sum of Square), αλλιώς SSE (Sum of Squared Errors), και βρίσκουμε το σημείο που η γραφική παράσταση κάνει «αγκώνα»
  - WCSS: το άθροισμα των τετραγώνων των αποστάσεων μεταξύ κάθε σημείου x και του μέσου μ της συστάδας στην οποία το x ανήκει

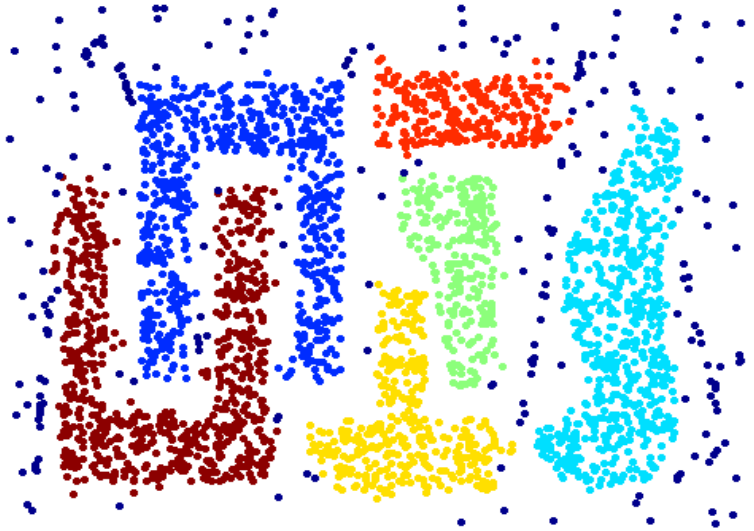
$$WCSS(K) = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$



πηγή εικόνας: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/

Συσταδοποίηση με βάση την πυκνότητα

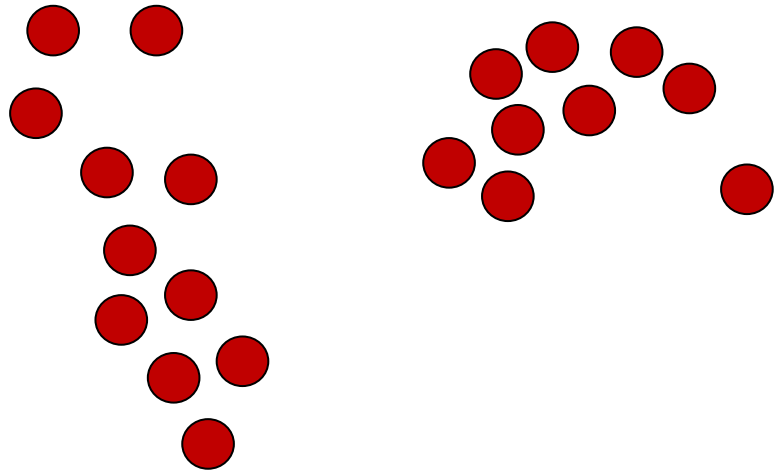
- Διπλός στόχος:
  - Τα γειτονικά σημεία να ενταχθούν στην ίδια συστάδα
  - Τα ακραία σημεία («θόρυβος») να απομονωθούν
- Οι πιο δημοφιλείς τεχνικές:
  - **DBSCAN – density-based spatial clustering of applications with noise** («Συσταδοποίηση βάσει πυκνότητας εφαρμογών με θόρυβο»)
  - **OPTICS -- ordering points to identify the clustering structure** («Διάταξη σημείων για την ταυτοποίηση της δομής των συστάδων»)

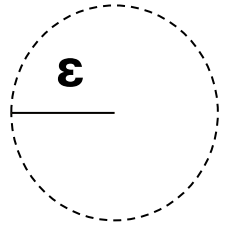


DBSCAN

- 2 παράμετροι εισόδου:
  - **MinPts** – κατώφλι πληθυσμού: ελάχιστος αριθμός σημείων μέσα στη συστάδα
  - **ε** – κατώφλι απόστασης: για κάθε σημείο της συστάδας θα πρέπει να υπάρχει ένα άλλο σημείο της συστάδας με απόσταση μικρότερη από ε.

  (Δηλαδή, δεν απαιτείται να δοθεί ως είσοδος το πλήθος k των συστάδων)
- Έξοδος: συστάδες & θόρυβος

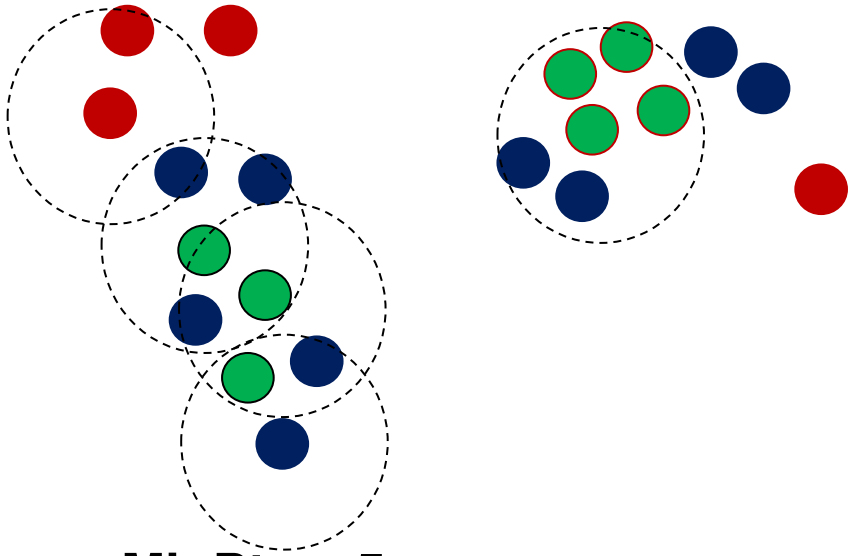




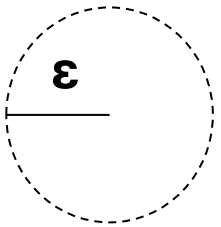
DBSCAN – η έννοια της πυκνότητας (1)

- **ε-γειτονιά** (**ε-neighborhood**) ενός σημείου p: σύνολο σημείων σε απόσταση ≤ ε από το p
- ένα σημείο p με πληθυσμό ε-γειτονιάς (συμπεριλαμβανομένου του p) ≥ MinPts ονομάζεται **πυρήνας** (**core**)
- ένα σημείο p που δεν είναι πυρήνας αλλά ανήκει στην ε-γειτονιά ενός πυρήνα ονομάζεται **σύνορο** (**border**)
- ένα σημείο p που δεν είναι ούτε πυρήνας ούτε σύνορο ονομάζεται **θόρυβος** (**noise**)

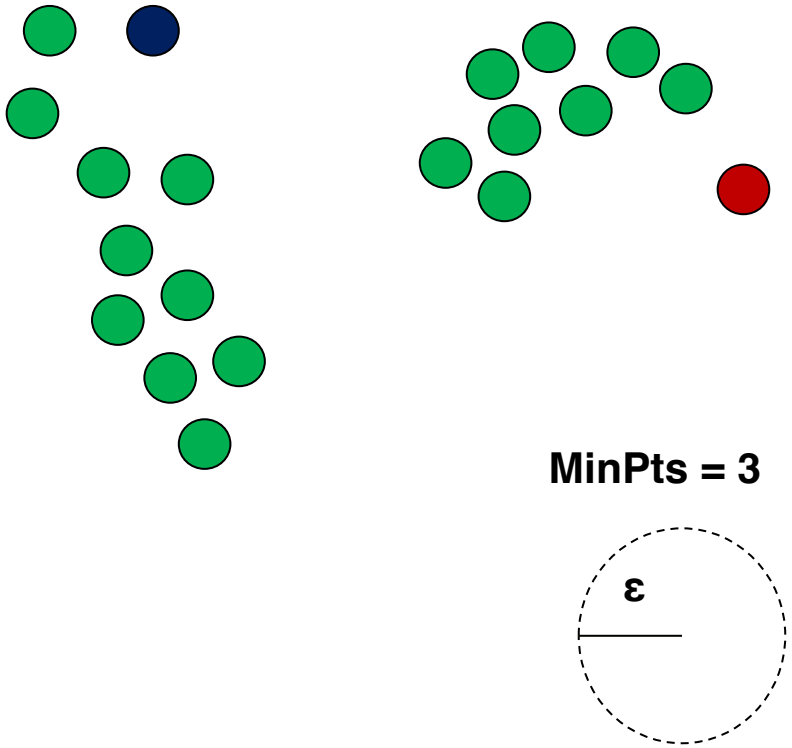
- κεντρική ιδέα αλγορίθμου:
  - να σχηματιστούν συστάδες γύρω από πυρήνες,
  - τα σύνορα να ενσωματωθούν σε αυτές,
  - ο θόρυβος να απομονωθεί



MinPts = 5
➔ 2 clusters, 4 outliers

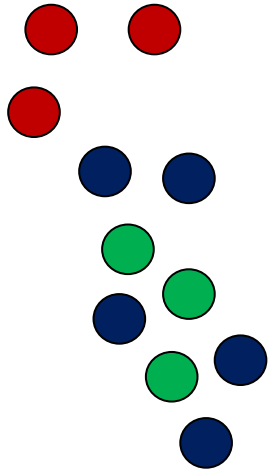


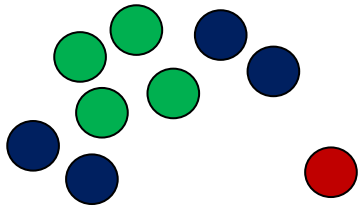
DBSCAN – η έννοια της πυκνότητας (2)



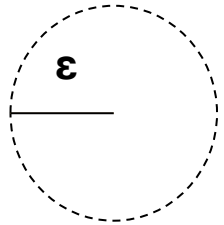
MinPts = 3

Output: 2 clusters + 1 outlier point

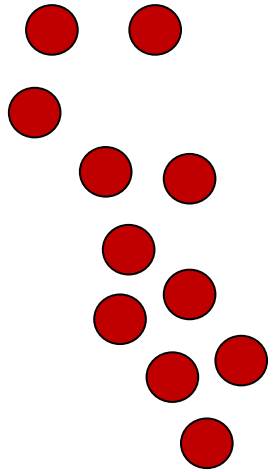


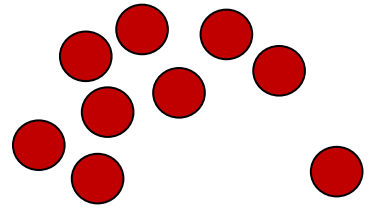


MinPts = 5

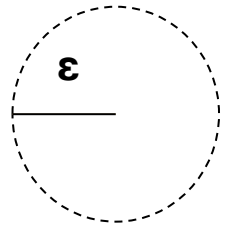


Output: 2 clusters + 4 outlier points



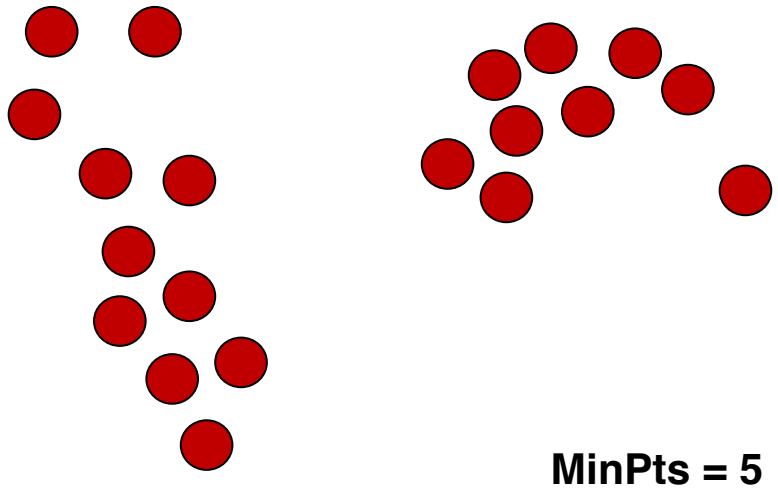


MinPts = 7



Output: 0 clusters + 20 outlier points

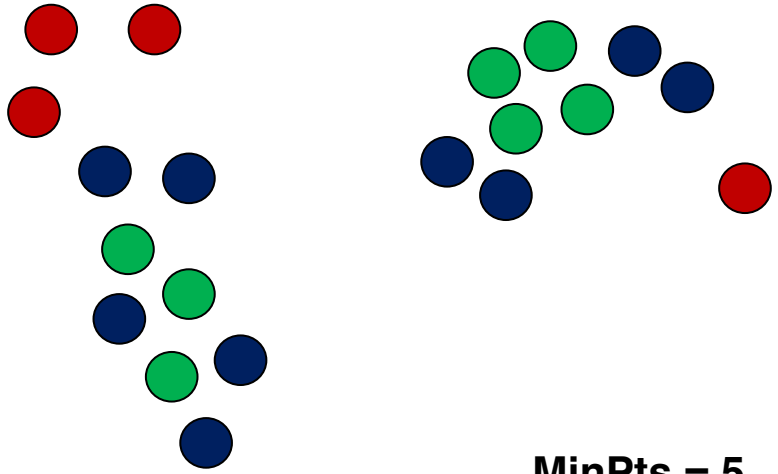
DBSCAN – η έννοια της πυκνότητας (3)



MinPts = 5

ε/2

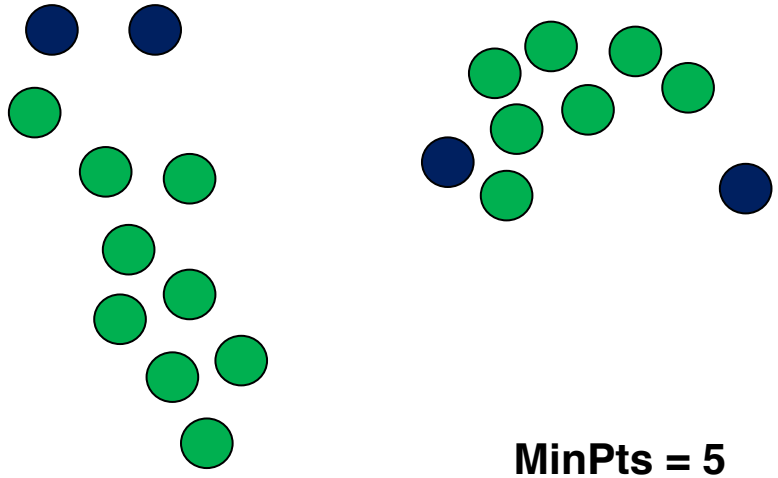
Output: 0 clusters +
20 outlier points



MinPts = 5

ε

Output: 2 clusters +
4 outlier points



MinPts = 5

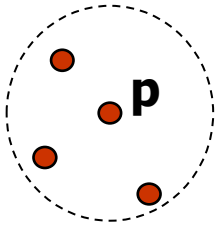
3ε/2

Output: 2 clusters +
0 outlier points

DBSCAN – η έννοια της συνδεσιμότητας (1)

- Ορισμοί των εννοιών:
  - **γειτονιά** (neighborhood), **απευθείας προσεγγισιμότητα βάσει πυκνότητας** (directly density-reachability), **προσεγγισιμότητα βάσει πυκνότητας** (density-reachability), **συνδεσιμότητα βάσει πυκνότητας** (density-connectivity)

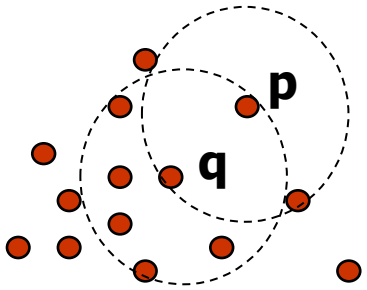
  … με δεδομένες τιμές των παραμέτρων ε, MinPts

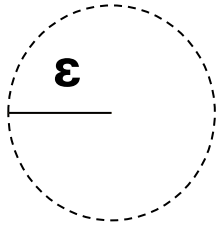


- **γειτονιά** σημείου p:
  - σύνολο σημείων σε απόσταση μέχρι ε από το σημείο p, δηλ. $N_\varepsilon(p)$: {q ανήκει στο D | dist(p,q) ≤ ε}

MinPts = 5

- σημείο p είναι **απευθείας προσεγγίσιμο βάσει πυκνότητας από** σημείο q αν:
  - p βρίσκεται στη γειτονιά του q, δηλ. $p \in N_\varepsilon(q)$, και
  - q είναι πυρήνας, δηλ. $|N_\varepsilon(q)| \geq$ MinPts

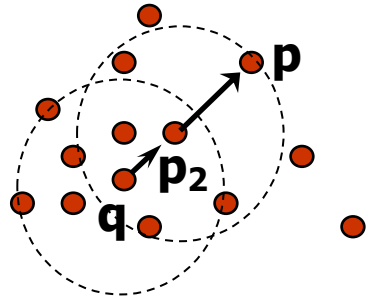




DBSCAN – η έννοια της συνδεσιμότητας (2)

- Ορισμοί των εννοιών:
  - **γειτονιά** (neighborhood), **απευθείας προσεγγισιμότητα βάσει πυκνότητας** (directly density-reachability), **προσεγγισιμότητα βάσει πυκνότητας** (density-reachability), **συνδεσιμότητα βάσει πυκνότητας** (density-connectivity)

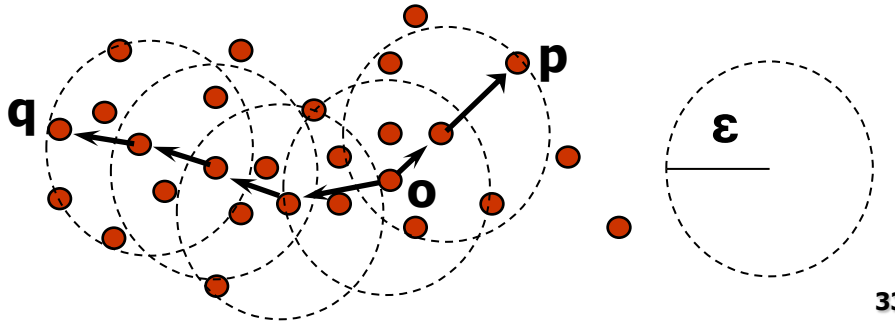
  … με δεδομένες τιμές των παραμέτρων ε, MinPts



- σημείο p είναι **προσεγγίσιμο βάσει πυκνότητας από** σημείο q αν:
  - υπάρχει αλυσίδα σημείων p_1, …, p_n, όπου p_1 = q, p_n = p, τέτοια ώστε: p_{i+1} απευθείας προσεγγίσιμο από p_i

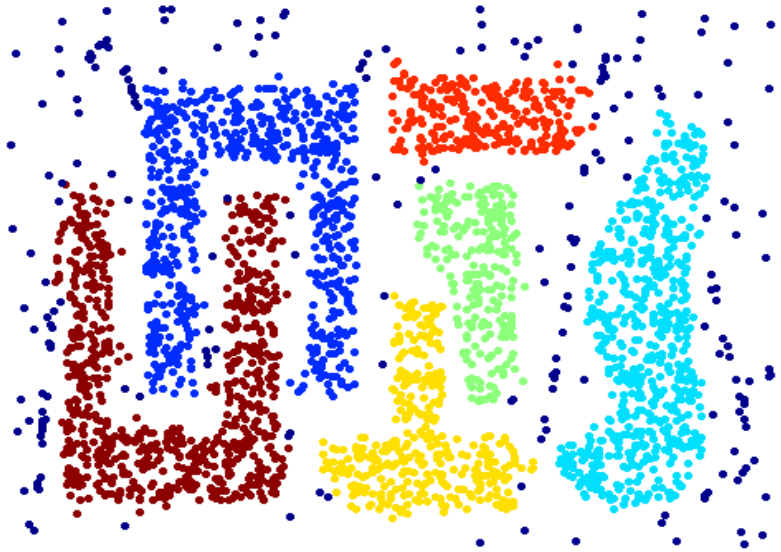
MinPts = 5

- σημεία p και q είναι **συνδεδεμένα βάσει πυκνότητας** αν:
  - υπάρχει σημείο o τέτοιο ώστε: p προσεγγίσιμο από o και q προσεγγίσιμο από o



DBSCAN – ο αλγόριθμος (1)

- Μια συστάδα ορίζεται ως το **μέγιστο σύνολο συνδεδεμένων σημείων** (με βάση τον προηγούμενο ορισμό της συνδεσιμότητας)
- Φορμαλιστικά, μία συστάδα C ικανοποιεί 2 κριτήρια:
  1. **Κριτήριο μεγιστότητας (Maximality)**:
     ∀ p ∈ C, q, εάν q είναι προσεγγίσιμο βάσει πυκνότητας από p, τότε q ∈ C
  2. **Κριτήριο συνδεσιμότητας (Connectivity)**:
     ∀ p, q ∈ C, p και q είναι συνδεδεμένα βάσει πυκνότητας



Συστάδες & Θόρυβος

- Άρα ο αλγόριθμος είναι σε θέση να ανακαλύπτει συστάδες (διαφόρων μεγεθών & σχημάτων), καθώς και θόρυβο

DBSCAN – ο αλγόριθμος (2)

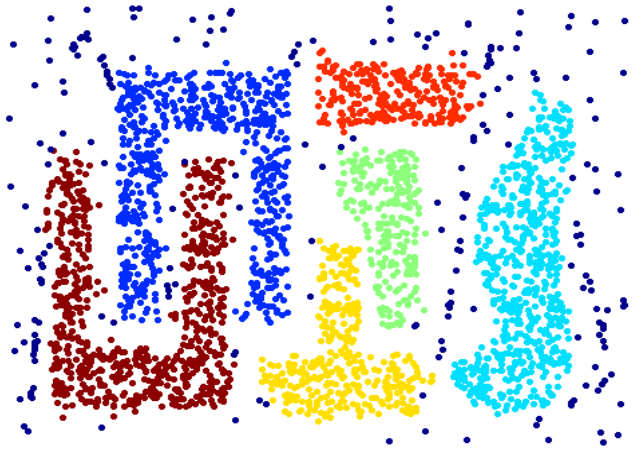
Πολυπλοκότητα: $O(n^2)$ ή $O(n \log n)$ αν υπάρχει χωρικό ευρετήριο (index), π.χ. R-tree, για την επιτάχυνση των ερωτημάτων χωρικής γειτνίασης

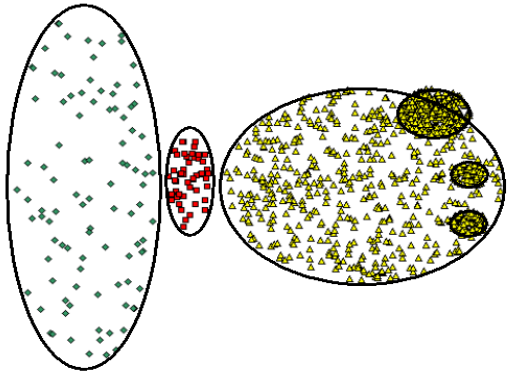
```
DBSCAN(DB, distFunc, eps, minPts) {
    C = 0                                                   /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop
*/
        Neighbors N = RangeQuery(DB, distFunc, P, eps)      /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) = Noise                                /* Label as Noise */
            continue
        }
        C = C + 1                                           /* next cluster label */
        label(P) = C                                        /* Label initial point */
        Seed set S = N \ {P}                                /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point */
            if label(Q) = Noise then label(Q) = C           /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed */
            label(Q) = C                                    /* Label neighbor */
            Neighbors N = RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check */
                S = S ∪ N                                   /* Add new neighbors to seed set */
            }
        }
    }
}
```

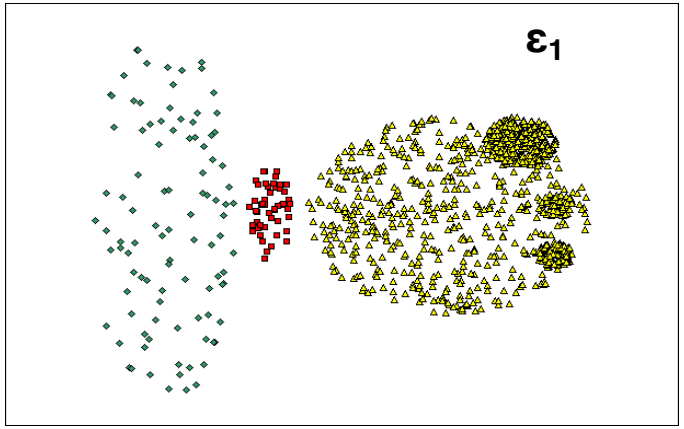
Πηγή: Wikipedia.org

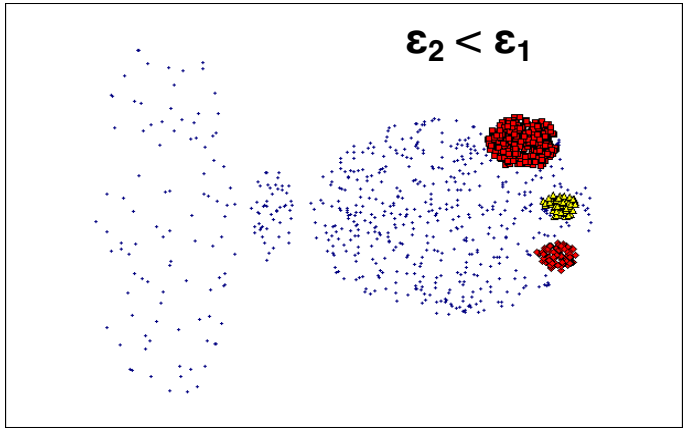
Υπέρ και κατά του DBSCAN

- Απομόνωση του θορύβου
- Ανακάλυψη συστάδων διαφορετικών μεγεθών / σχημάτων
- Ευαισθησία στις εναλλαγές στην πυκνότητα του συνόλου δεδομένων



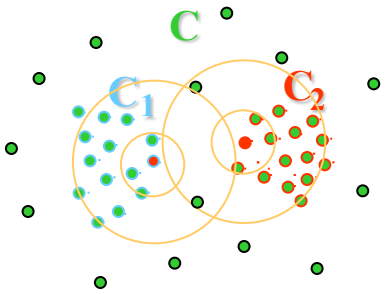




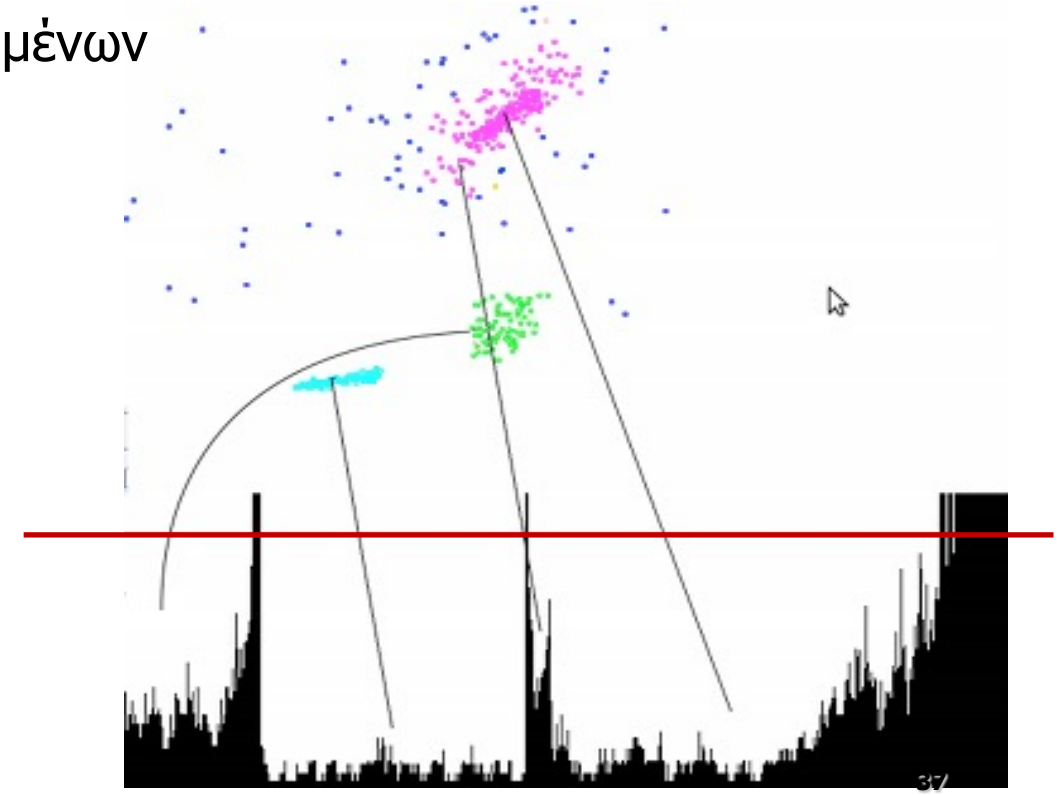


DBSCAN visualization: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

OPTICS (1)



- Στόχος: η αντιμετώπιση της κύριας αδυναμίας του DBSCAN (ευαισθησία στην εναλλαγή πυκνότητας)
- Παρατήρηση: οι περισσότερο πυκνές συστάδες εμπεριέχονται σε λιγότερο πυκνές συστάδες
- Ιδέα: γραμμική διάταξη των σημείων του συνόλου δεδομένων
  - Στη γραμμική διάταξη, η απόσταση μεταξύ 2 σημείων αντιστοιχεί στην πυκνότητα που πρέπει να ισχύει ώστε αυτά τα σημεία να τοποθετηθούν στην ίδια συστάδα
  - **Γραφική προσεγγισιμότητας** (reachability plot)
- Παράμετροι: ε, MinPts (όπως στον αλγόριθμο DBSCAN)
  - Η παράμετρος ε θα μπορούσε και να απουσιάζει – υπάρχει για πρακτικούς λόγους (απόδοσης)

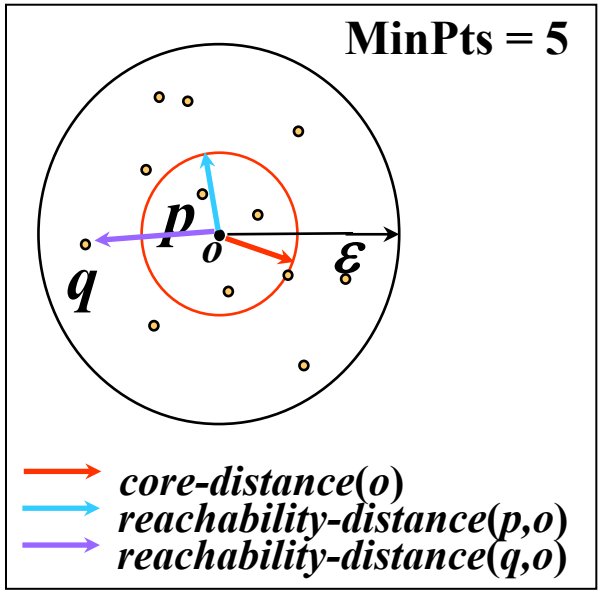


OPTICS (2)

$$\text{core-distance}_{\varepsilon,MinPts}(p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \text{distance to the } MinPts\text{-th closest point} & \text{otherwise} \end{cases}$$

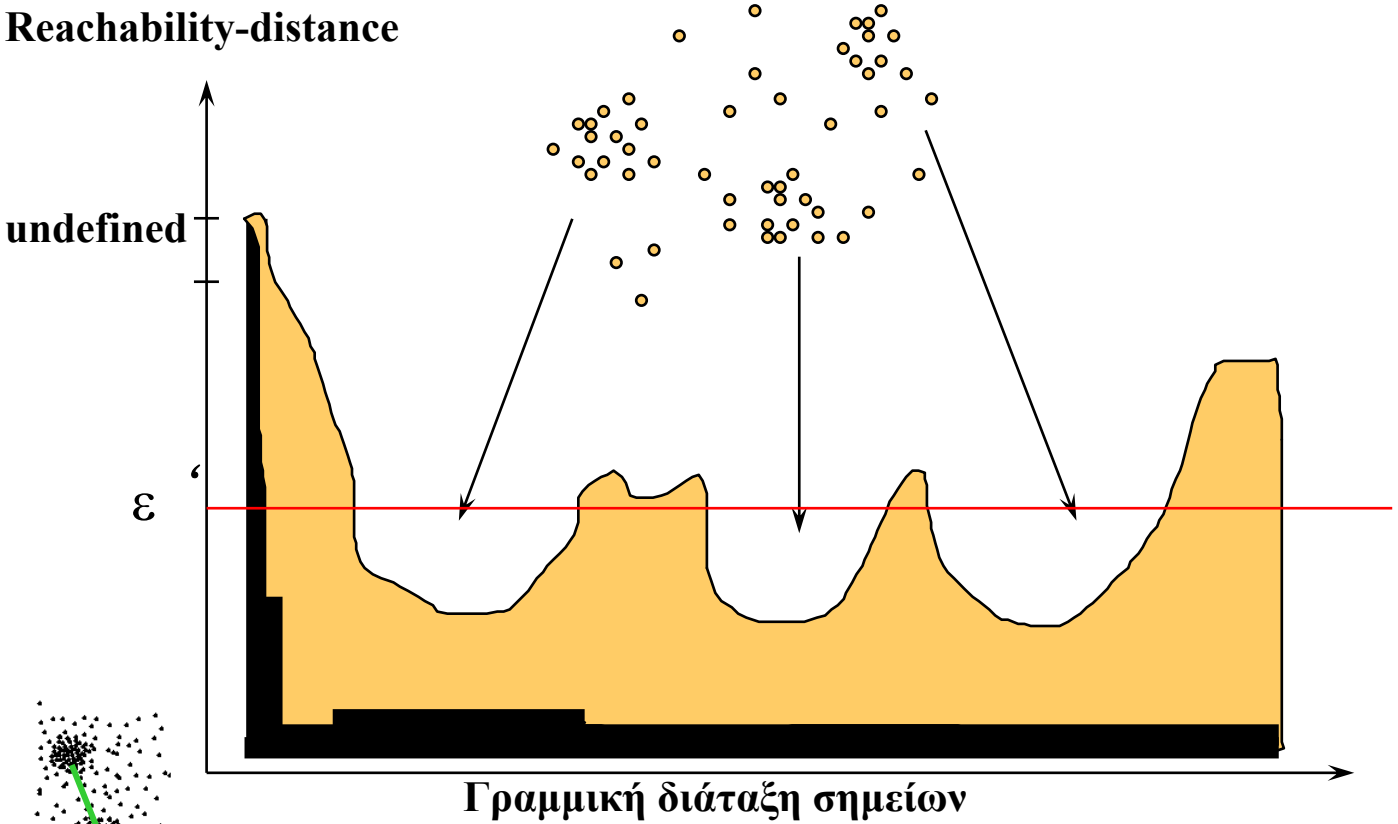
$$\text{reachability-distance}_{\varepsilon,MinPts}(o,p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \max(\text{core-distance}_{\varepsilon,MinPts}(p), \text{distance}(p,o)) & \text{otherwise} \end{cases}$$

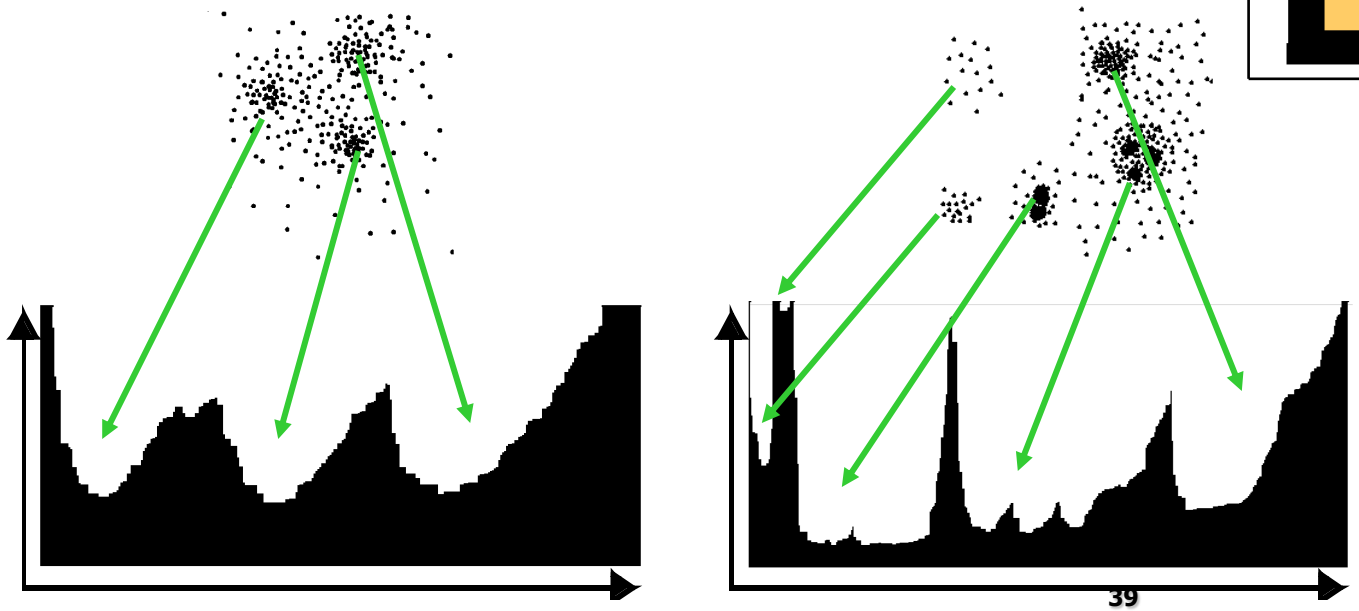
- **Απόσταση-πυρήνα (core-distance)** ενός σημείου p: η μικρότερη απόσταση ε ώστε η ε-γειτονιά του σημείου p να περιέχει τουλάχιστον MinPts σημεία
  - Με άλλα λόγια, η απόσταση του p από το MinPts-ο πλησιέστερο σημείο
- **Απόσταση-προσεγγισιμότητας (reachability-distance)** ενός σημείου ο από ένα σημείο p: η μέγιστη μεταξύ δύο παρακάτω αποστάσεων:
  - της απόστασης μεταξύ των 2 σημείων και
  - της απόστασης-πυρήνα του σημείου p (το οποίο p πρέπει να είναι σημείο-πυρήνας)



OPTICS (3)

- **Γραφική προσεγγισιμότητας** (reachability plot): διάταξη με βάση την απόσταση-προσεγγισιμότητας (reachability distance)
  - «**κοιλάδες**» (valleys) → συστάδες
  - «**λόφοι**» (hills) → θόρυβος





- Στα υπέρ του OPTICS: αυτόματη αλλά και διαδραστική χρήση

OPTICS (4)

Πολυπλοκότητα: $O(n^2)$ ή $O(n \log n)$ αν υπάρχει χωρικό ευρετήριο (index) για την επιτάχυνση των ερωτημάτων χωρικής γειτνίασης

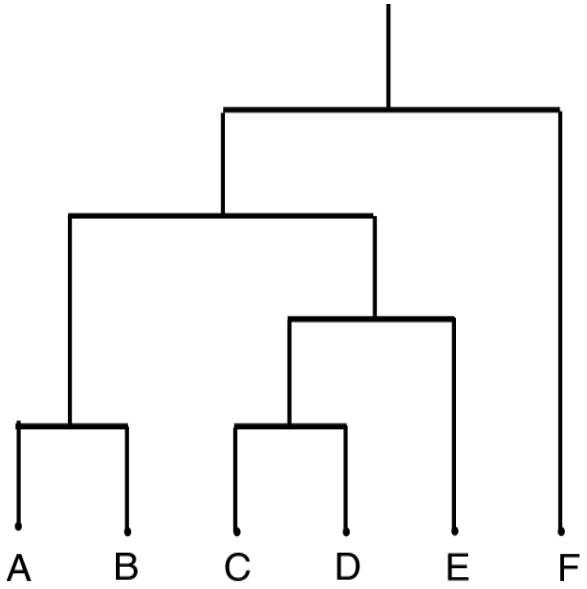
```
function OPTICS(DB, eps, MinPts) is
    for each point p of DB do
        p.reachability-distance = UNDEFINED
    for each unprocessed point p of DB do
        N = getNeighbors(p, eps)
        mark p as processed
        output p to the ordered list
        if core-distance(p, eps, MinPts) != UNDEFINED then
            Seeds = empty priority queue
            update(N, p, Seeds, eps, MinPts)
            for each next q in Seeds do
                N' = getNeighbors(q, eps)
                mark q as processed
                output q to the ordered list
                if core-distance(q, eps, MinPts) != UNDEFINED do
                    update(N', q, Seeds, eps, MinPts)
```

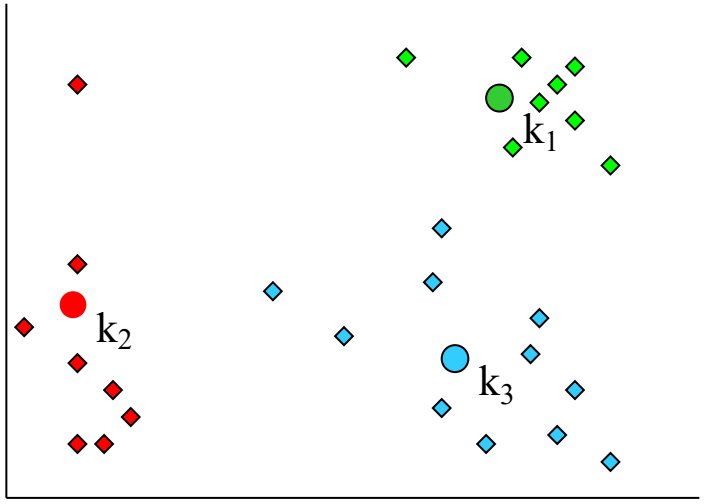
```
function update(N, p, Seeds, eps, MinPts) is
    coredist = core-distance(p, eps, MinPts)
    for each o in N
        if o is not processed then
            new-reach-dist = max(coredist, dist(p,o))
            if o.reachability-distance == UNDEFINED then // o is not in Seeds
                o.reachability-distance = new-reach-dist
                Seeds.insert(o, new-reach-dist)
            else               // o in Seeds, check for improvement
                if new-reach-dist < o.reachability-distance then
                    o.reachability-distance = new-reach-dist
                    Seeds.move-up(o, new-reach-dist)
```

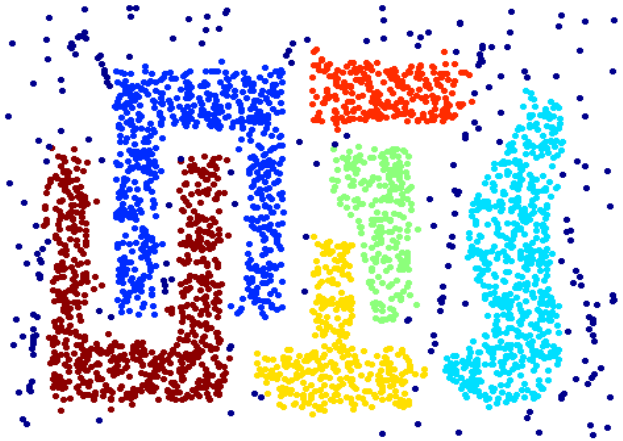
Πηγή: Wikipedia.org

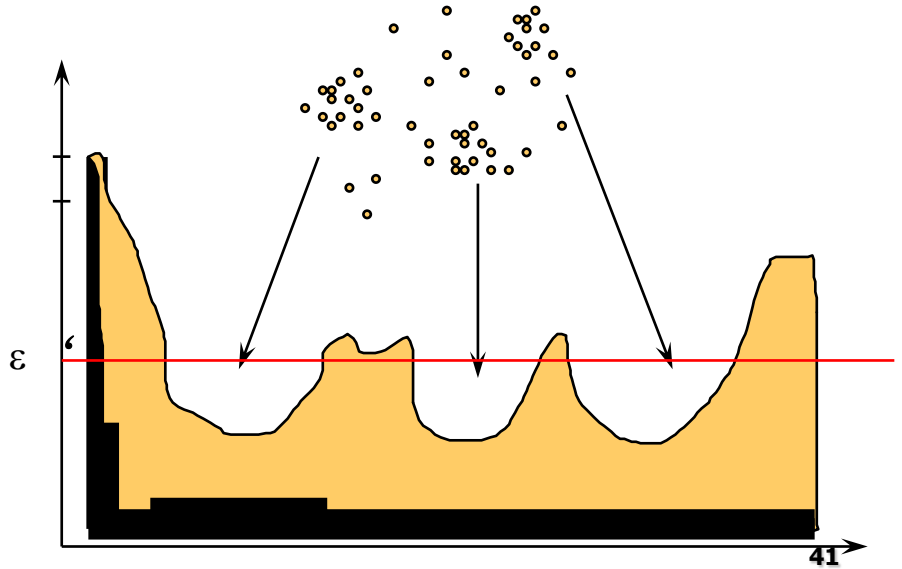
Σύνοψη

- Συσταδοποίηση: η εύρεση ομάδων μεταξύ των δεδομένων ενός συνόλου
- Μεγάλη ποικιλία τεχνικών









Για περαιτέρω μελέτη

- Βιβλιογραφία (κατά σειρά: η μέθοδος k-means και μια κριτική στο «κριτήριο του αγκώνα», οι μέθοδοι DBSCAN και OPTICS, καθώς και 2 άρθρα επισκόπησης)
  - MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proc 5th Berkeley Symp Math Stat Probab 1: 281–297.
  - Schubert E (2022) Stop using the elbow criterion for k-means and how to choose the number of clusters instead. ACM SIGKDD Explorations Newsletter, 25(1): 36–42.
  - Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc ACM SIGKDD Int Conf Knowledge Discovery and Data Mining (KDD), pp. 226–231.
  - Ankerst M, Breunig M, Kriegel H, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: Proc ACM SIGMOD Int Conf Management of Data, pp. 49–60.
  - Anil KJ, Murty MN, Flynn PJ (1999) Data clustering: A review. ACM Comput. Surv. 31(3): 264-323.
  - Xu D, Tian YA (2015) Comprehensive survey of clustering algorithms. Ann. Data. Sci. 2: 165–193.
- Ανοιχτός κώδικας:
  - scikit-learn: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster
  - PyClustering library: https://codedocs.xyz/annoviko/pyclustering/index.html