Συσταδοποίηση ΙΙ

DBScan

Εγκυρότητα Συσταδοποίησης

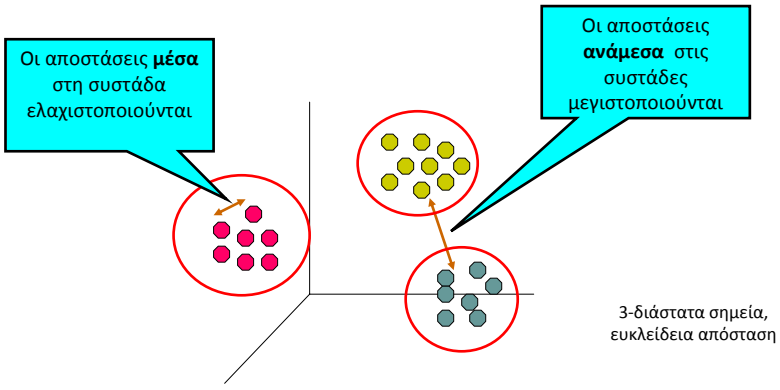
BIRCH

Μέρος των διαφανειών είναι από το **P.-N. Tan, M.Steinbach, V. Kumar, «Introduction to Data Mining»**, Addison Wesley, 2006

Επανάληψη

Τι είναι συσταδοποίηση

Εύρεση συστάδων αντικειμένων έτσι ώστε τα αντικείμενα σε κάθε συστάδα να είναι *όμοια (ή να σχετίζονται)* και *διαφορετικά (ή μη σχετιζόμενα)* από τα αντικείμενα των άλλων συστάδων



Είδη Συστάδων

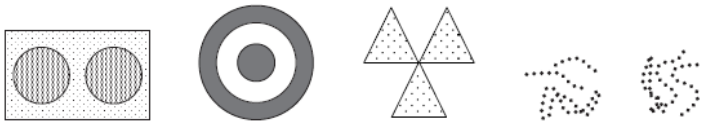
Συστάδες βασισμένες σε κέντρο ή πρότυπο Μια συστάδα είναι ένα σύνολο από αντικείμενα τέτοιο ώστε ένα αντικείμενο στην ομάδα είναι κοντινότερο σε (ή πιο όμοιο με) το «κέντρο» (κεντρικό σημείο ή κέντρο βάρους) ή πρότυπο (medoid) της ομάδας από ότι από το κέντρο οποιασδήποτε άλλης ομάδας.

Συνεχής συστάδες (Contiguous Cluster): μια συστάδα είναι ένα σύνολο σημείων τέτοιο ώστε κάθε σημείο είναι πιο κοντά σε ένα ή περισσότερα σημεία της συστάδας από ό,τι σε οποιοδήποτε σημείο εκτός συστάδας

Συστάδες βασισμένες στην πυκνότητα: μια συστάδα είναι μια πυκνή περιοχή από σημεία την οποία χωρίζουν από άλλες περιοχές μεγάλης πυκνότητας περιοχές χαμηλής πυκνότητας

Είδη Συστάδων

Παράδειγμα



- βάση κέντρου
- συνεχής
- βάση πυκνότητας

Είδη Συστάδων

- Διαχωριστικές (Partitioning) Μέθοδοι
  - κατασκευάζουν διαχωρισμούς του χώρου και τους βελτιώνουν επαναληπτικά
- Ιεραρχικές Μέθοδοι

K-means: Βασικός Αλγόριθμος

- Διαχωριστική μέθοδος
- Κατασκευάζει Κ συστάδες (είσοδος στο πρόβλημα)
- Κάθε συστάδα συσχετίζεται με κάποιο σημείο- centroid (κεντρικό σημείο)
- Κάθε σημείο συσχετίζεται με την κοντινότερη του από της Κ συστάδες (δηλ. κεντρικά σημεία)

Συνήθως Ευκλείδεια απόσταση

Βασικός αλγόριθμος

1: Επιλογή *K* σημείων ως τα αρχικά κεντρικά σημεία
2: **Repeat**
3: Ανάθεση όλων των αρχικών σημείων στο *κοντινότερο* τους από τα *K* κεντρικά σημεία
4: Επανα-υπολογισμός του *κεντρικού σημείου* κάθε συστάδας
5: **Until** τα κεντρικά σημεία να μην αλλάζουν

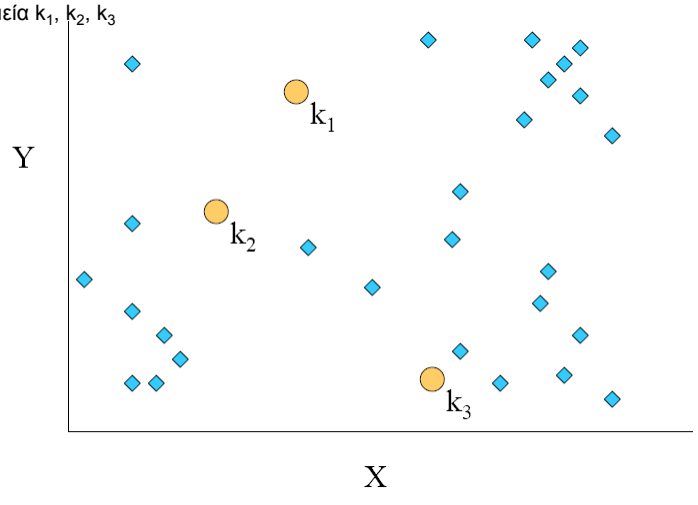
Μέσο (κέντρο βάρους)

K-means: Βασικός Αλγόριθμος

Αρχική κατάσταση,

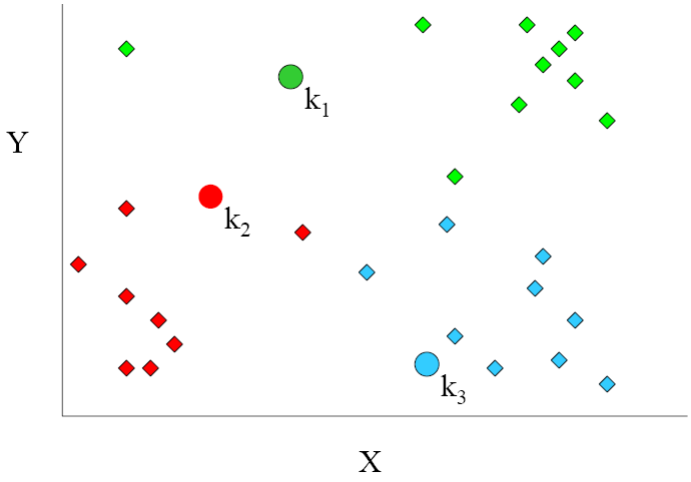
K = 3 συστάδες

Αρχικά σημεία k_1, k_2, k_3



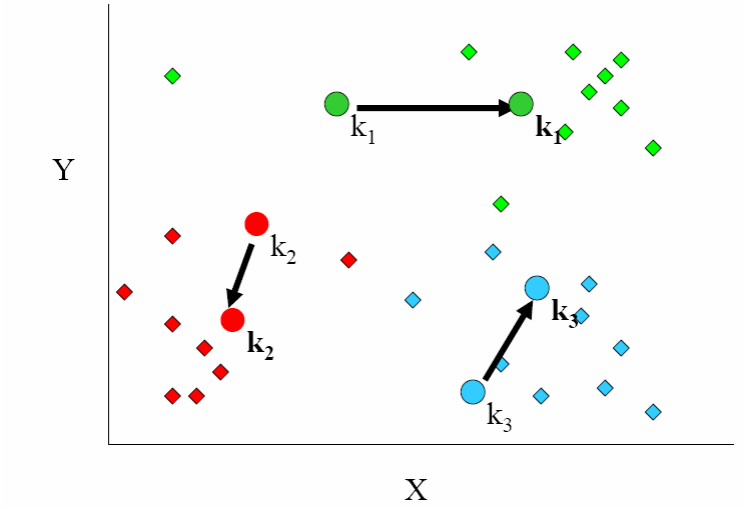
K-means: Βασικός Αλγόριθμος

Τα σημεία ανατίθενται στο πιο γειτονικό από τα 3 αρχικά σημεία



K-means: Βασικός Αλγόριθμος

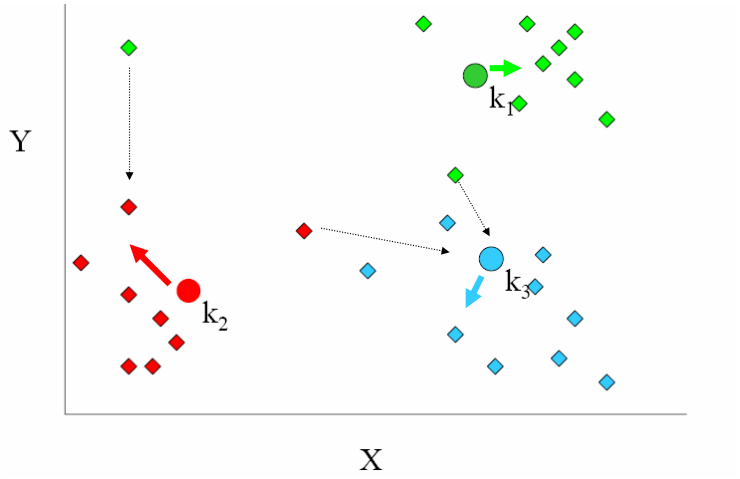
Επανα-υπολογισμός του κέντρου (κέντρου βάρους) κάθε σημείου



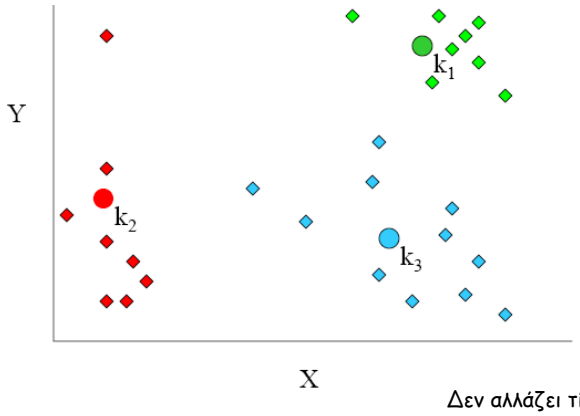
K-means: Βασικός Αλγόριθμος

Νέα ανάθεση των σημείων

Νέα κέντρα βάρους



K-means: Βασικός Αλγόριθμος



Δεν αλλάζει τίποτα -> ΤΕΛΟΣ

K-means: Βασικός Αλγόριθμος

Προβλήματα με:

- μη σφαιρικά σχήματα
- μέγεθος
- πυκνότητα
- ακραία σημεία (outliers)

Άθροισμα του Τετραγωνικού Σφάλματος (ΑΤΣ)

-- Sum of Squared Error (SSE)

Για όλες τις Κ συστάδες

$$SSE(\text{ΑΤΣ}) = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

Τετραγωνικό λάθος, για κάθε συστάδα C_i

Για όλα τα σημεία $x \in C_i$, παίρνουμε την απόσταση τους από ένα αντιπροσωπευτικό σημείο (m_i) της συστάδας (το κέντρο βάρους για Ευκλείδειες αποστάσεις)

- Το σημείο που ελαχιστοποιεί το σφάλμα είναι το κέντρο βάρους κάθε πλειάδας

Άθροισμα Απόλυτου Σφάλματος (ΑΑΣ)

Για όλες τις Κ συστάδες

$$\text{ΑΑΣ} = \sum_{i=1}^{K} \sum_{x \in C_i} dist_{L1}(m_i, x)$$

Διαφορετικές συναρτήσεις σφάλματος, πχ Manhattan (L1)

- Το σημείο που ελαχιστοποιεί το σφάλμα είναι το (μεσαίο σημείο) διάμεσος

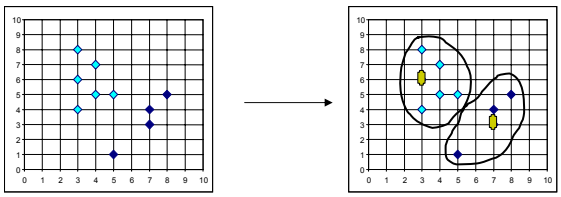
K-medoid

Αντί για το μέσο (που μπορεί να μην ανήκει στο αρχικό σύνολο σημείων) διαλέγει κάθε φορά ένα αντιπροσωπευτικό σημείο από τα δεδομένα και ελαχιστοποιεί την απόσταση από αυτό

Medoid: το πιο κεντρικό σημείο της συστάδας (αντί να χρησιμοποιεί το mean), το σημείο με τη μικρότερη μέση απόσταση από όλα τα σημεία της ομάδας

Μειώνει την ευαισθησία σε outliers

Μπορεί να εφαρμοστεί σε δεδομένα οποιουδήποτε τύπου (πχ και για κατηγορικά δεδομένα)

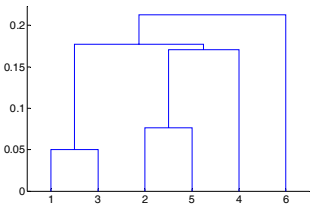


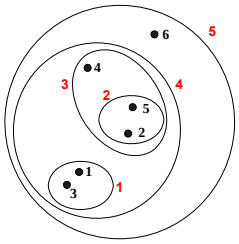
Ιεραρχική Συσταδοποίηση: Βασικά

Παράγει ένα σύνολο από εμφωλευμένες συστάδες οργανωμένες σε ένα ιεραρχικό δέντρο

Μπορεί να παρασταθεί με ένα δένδρο-γραμμα

Ένα διάγραμμα που μοιάζει με δένδρο και καταγράφει τις ακολουθίες από συγχωνεύσεις (merges) και διαχωρισμούς (splits)





Ιεραρχική Συσταδοποίηση

Δυο βασικοί τύποι ιεραρχικής συσταδοποίησης

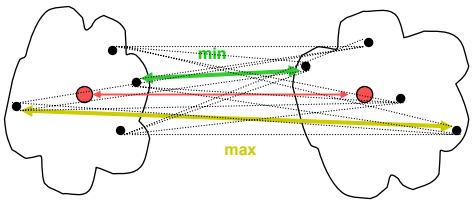
- Συσσωρευτικός (Agglomerative):
  - Αρχίζει με τα σημεία ως ξεχωριστές συστάδες
  - Σε κάθε βήμα, συγχωνεύει το πιο κοντινό ζευγάρι συστάδων μέχρι να μείνει μόνο μία (ή k) συστάδες
- Διαιρετικός (Divisive):
  - Αρχίζει με μία συστάδα που περιέχει όλα τα σημεία
  - Σε κάθε βήμα, διαχωρίζει μία συστάδα, έως κάθε συστάδα να περιέχει μόνο ένα σημείο (ή να δημιουργηθούν k συστάδες)

Συσσωρευτική Ιεραρχική Συσταδοποίηση (ΣΙΣ)

Βασικός Αλγόριθμος

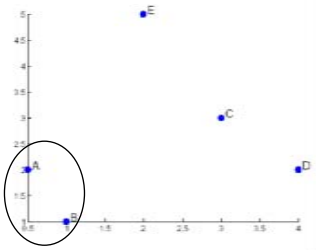
1: Υπολογισμός του Πίνακα Γειτνίασης
2: Έστω κάθε σημείο αποτελεί και μια συστάδα
3: **Repeat**
4: Συγχώνευση των δύο κοντινότερων συστάδων
5: Ενημέρωση του Πίνακα Γειτνίασης
6: **Until** να μείνει μία μόνο συστάδα

ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων



- MIN (μοναδικής ακμής) single link
- MAX (πλήρους συνδεσιμότητας)
- Μέσος όρος των αποστάσεων των σημείων των συστάδων
- Η απόσταση μεταξύ των κεντρικών σημείων (κέντρου βάρους)
- Άλλες μέθοδοι βασισμένες σε μια αντικειμενική συνάρτηση
  - Η μέθοδος του Ward χρησιμοποιεί τετραγωνικά λάθη

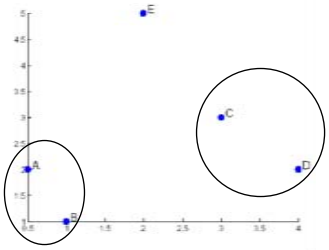
ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων



	A	B	C	D	E
A	0	1.1180 [1]	2.6926	3.5	3.3541
B	1.1180	0	2.8282	3.1623	4.1231
C	2.6926	2.8284	0	1.4142	2.2361
D	3.5	3.1623	1.4142	0	3.6056
E	3.3541	4.1231	2.2361	3.6056	0

πίνακας γειτνίασης

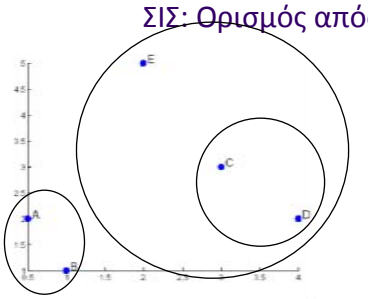
ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων



	A	B	C	D	E
A	0	1.1180 [1]	2.6926	3.5	3.3541
B	1.1180	0	2.8282	3.1623	4.1231
C	2.6926	2.8284	0	1.4142 [2]	2.2361
D	3.5	3.1623	1.4142	0	3.6056
E	3.3541	4.1231	2.2361	3.6056	0

πίνακας γειτνίασης

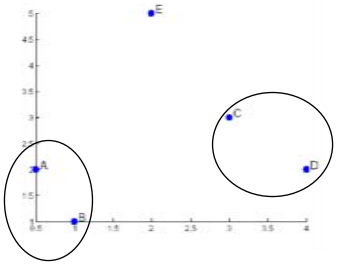
ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων



	A	B	C	D	E
A	0	1.1180 (1)	2.6926	3.5	3.3541
B	1.1180	0	2.8282	3.1623 (4)	4.1231
C	2.6926	2.8284	0	1.4142 (2)	2.2361 (3)
D	3.5	3.1623	1.4142	0	3.6056
E	3.3541	4.1231	2.2361	3.6056	0

πίνακας γειτνίασης

ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων



	A	B	C	D	E
A	0	1.1180 (1)	2.6926	3.5	3.3541
B	1.1180	0	2.8282	3.1623	4.1231
C	2.6926	2.8284	0	1.4142	2.2361
D	3.5	3.1623	1.4142	0	3.6056
E	3.3541	4.1231	2.2361	3.6056	0

πίνακας γειτνίασης

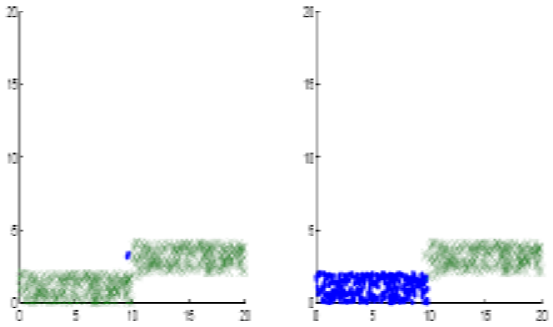
Δεν αρκεί να δούμε ένα link (μία απόσταση)

d(E, {C, D}) 2 αποστάσεις

d(E, {A, B}) 2 αποστάσεις

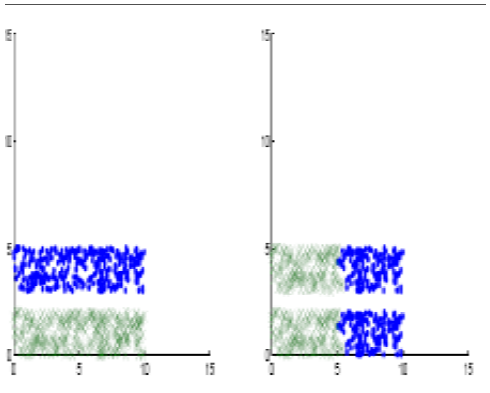
d({A, B}, {C, D}) 4 αποστάσεις

ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων: MAX



MIN MAX

ΣΙΣ: Ορισμός απόστασης μεταξύ συστάδων: MAX



MIN MAX

ΣΙΣ: Περιορισμοί και Προβλήματα

Οι αποφάσεις είναι τελικές – αφού δυο συστάδες συγχωνευτούν αυτό δεν μπορεί να αλλάξει

Δεν ελαχιστοποιούν άμεσα κάποια αντικειμενική συνάρτηση

DBSCAN

DBSCAN: Γενικά

Γειτονιά ενός σημείου p = όλα τα σημεία σε απόσταση Eps από το p:
NEps(p) = {q | dist(p,q) <= Eps }

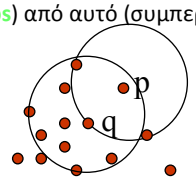
Δύο παράμετροι:

- **Eps** : Μέγιστη ακτίνα της γειτονιάς
- **MinPts**: Ελάχιστος αριθμός σημείων στην Eps-γειτονιά ενός σημείου

Ο DBSCAN είναι ένας αλγόριθμος βασισμένος στην πυκνότητα
Πυκνότητα για ένα σημείο = αριθμός σημείων (MinPts) μέσα σε μια προκαθορισμένη ακτίνα (Eps) από αυτό (συμπεριλαμβανομένου του σημείου)

Για το p έχουνε 4

Για το q έχουμε >5



MinPts = 5

ε = 1 cm

DBSCAN: Γενικά

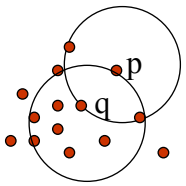
Τα σημεία διαχωρίζονται σε:

- Βασικά (core) – σημεία πυρήνα: ένα σημείο για το οποίο υπάρχουν περισσότερα από ένα προκαθορισμένο αριθμό (***MinPts***) σημεία σε ακτίνα ***Eps***

Αυτά είναι τα σημεία που είναι στο εσωτερικό μιας συστάδας (ομάδας πυκνών σημείων)

- Οριακά (border) – σημεία ορίου: ένα σημείο για το οποίο υπάρχουν λιγότερα από ένα προκαθορισμένο αριθμό (MinPts) σημεία σε ακτίνα Eps, αλλά είναι στη γειτονιά (τουλάχιστον) ενός βασικού σημείου

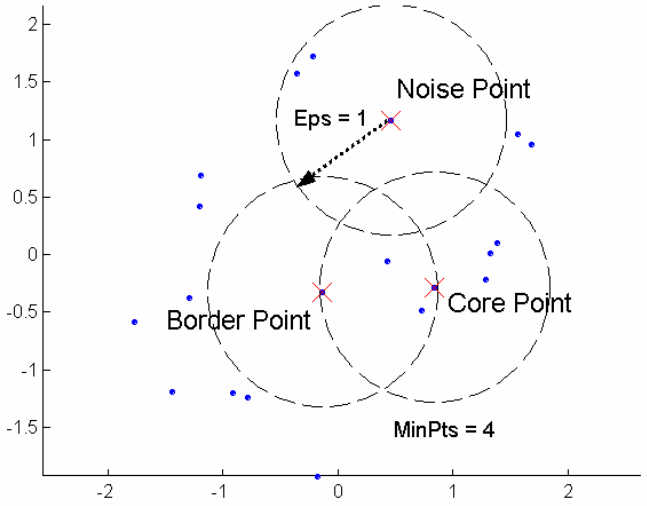
- Θορύβου (noise): ένα σημείο που δεν είναι ούτε σημείο πυρήνα ούτε σημείο ορίου



MinPts = 5

ε = 1 cm

DBSCAN: Γενικά



DBSCAN: Αλγόριθμος

Βασικός Αλγόριθμος

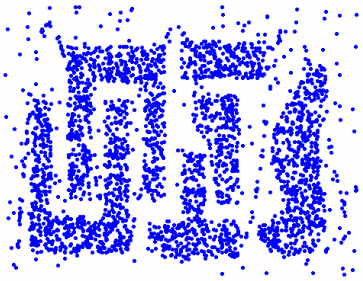
1: Χαρακτήρισε κάθε σημείο ως **πυρήνα**, **ορίου** ή **θορύβου**
2: Διέγραψε τα σημεία θορύβου

3: Τοποθέτησε μια ακμή μεταξύ όλων των σημείων πυρήνα που είναι σε απόσταση έως *Eps* μεταξύ τους
4: Κάνε κάθε ομάδα συνδεδεμένων σημείων πυρήνα μια διαφορετική συστάδα

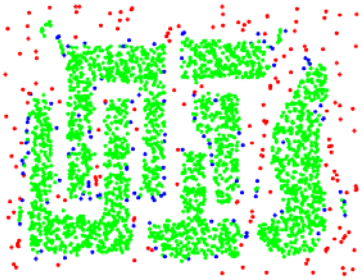
5: Ανάθεσε κάθε σημείο ορίου σε μία από τις συστάδες των συσχετιζόμενων του σημείων πυρήνα

DBSCAN: Αλγόριθμος

Βήμα 1&2



Αρχικά σημεία

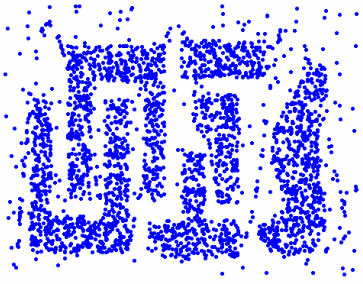


Τύποι σημείων: core, border και noise

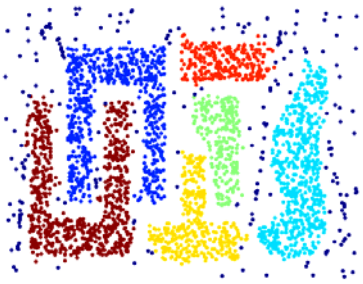
Eps = 10, MinPts = 4

DBSCAN: Πλεονεκτήματα

Βήμα 3&4



Αρχικά Σημεία



Συστάδες

- Δεν επηρεάζεται από το θόρυβο
- Μπορεί να χειριστεί συστάδες με διαφορετικά σχήματα και μεγέθη

DBSCAN: Πολυπλοκότητα

Για m σημεία εισόδου:

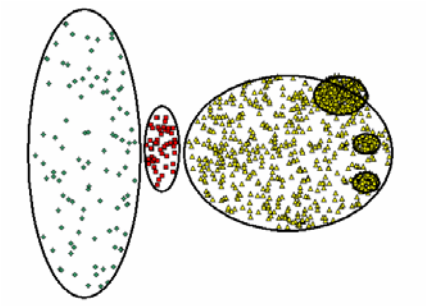
O(m x χρόνος εντοπισμού σημείων σε eps-γειτονιά)

O(m^2)

Για μικρό αριθμό διαστάσεων, υπάρχουν δομές που υποστηρίζουν την πράξη σε *O*(m logm)

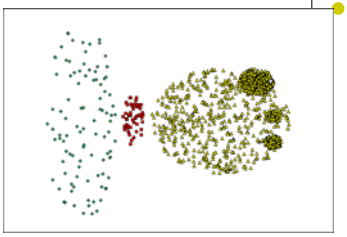
O(m) χώρος (για κάθε σημείο κρατάμε μόνο ένα label σε μια συστάδα ανήκει και το είδος του (βασικό, οριακό, θόρυβος))

DBSCAN: Περιορισμοί

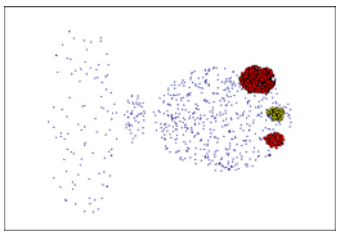


Αρχικά Σημεία

- Διαφορετικές πυκνότητες
- Πολυ-διάστατα δεδομένα – δύσκολος ορισμός πυκνότητας και δαπανηρός υπολογισμός γειτόνων



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

DBSCAN: Καθορισμός των MinPts και Eps

Η ιδέα είναι να κοιτάξουμε την απόσταση ενός σημείου από τον k-οστό κοντινότερο γείτονα του -> k-dist

Γενικά (κατά μέσο όρο), για τα σημεία που ανήκουν στην ίδια ομάδα, η τιμή του k-dist θα είναι μικρή (αν το k δεν είναι μεγαλύτερο από το μέγεθος της συστάδας)

Θα θέλαμε για τα σημεία μιας συστάδας, να έχουν περίπου την ίδια k-dist

Τα σημεία θορύβου έχουν μεγαλύτερες k-dist

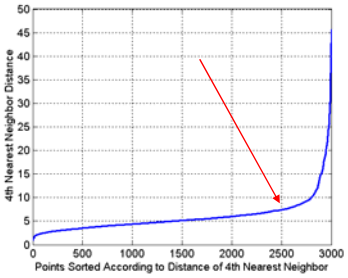
1. Υπολογίζουμε την k-dist για όλα τα σημεία, για κάποιο k
2. Ταξινομούμε τις αποστάσεις με φθίνουσα διάταξη

Περιμένουμε ξαφνική αλλαγή στο k-dist που αντιστοιχεί στο Eps

Οπότε k = MinPts και Eps = k-dist

Eps ~ 7

MinPts = 4



DBSCAN: Γενικά

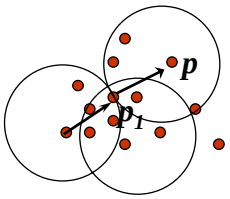
Συσταδοποίηση βασισμένη στην πυκνότητα (τοπικό κριτήριο)

Βασικά χαρακτηριστικά:

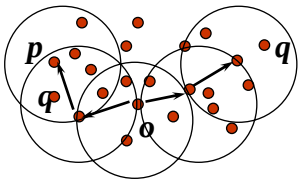
- Ανακαλύπτουν συστάδες οποιουδήποτε σχήματος
- Αντιμετωπίζουν το θόρυβο
- Μία διάσχιση (scan) των δεδομένων
- Χρειάζονται παραμέτρους εισόδου για την πυκνότητα
- Δύσκολο να ανακαλύψουν συστάδες με διαφορετική πυκνότητα
- Στις πολλές διαστάσεις, η έννοια της πυκνότητας είναι ασαφής
- Το κόστος εξαρτάται από το κόστος υπολογισμού του κοντινότερου γείτονα

DBSCAN: Τυπικός ορισμός

- Density-reachable (προσπελάσιμο με βάση τη πυκνότητα):
  - Ένα σημείο *p* είναι density-reachable από ένα σημείο *q* αν υπάρχει μια αλυσίδα από σημεία $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ τέτοια ώστε το p_{i+1} να είναι στη γειτονιά του p_i



- Density-connected
  - Ένα σημείο *p* είναι density-connected σε ένα σημεία *q* αν υπάρχει ένα σημείο *o* τέτοιο ώστε και το *p* and *q* να είναι density-reachable από το *o*



- Συστάδα είναι το μέγιστο (maximal) σύνολο από density-connected σημεία

Εγκυρότητα Συσταδοποίησης
Cluster validity

Ποιότητα Συσταδοποίησης

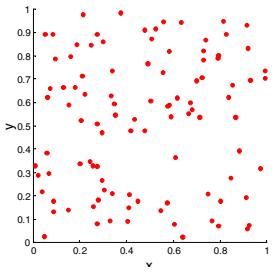
Ποιότητα ή εγκυρότητα συσταδοποίησης: Πόσο καλή είναι η συσταδοποίηση που επιτύχαμε;

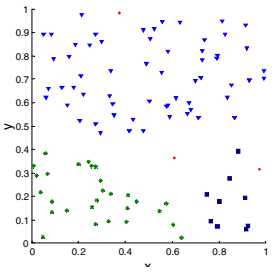
Οι αλγόριθμοι που είδαμε παράγουν κάποιες συστάδες ακόμα και όταν τα δεδομένα παράγονται τυχαία

Δύσκολη η αξιολόγηση, ιδιαίτερα σε πολλές διαστάσεις

Συστάδες σε Τυχαία Δεδομένα

Τυχαία Σημεία

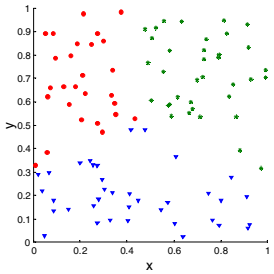


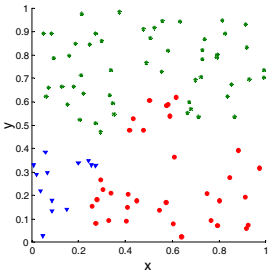


DBSCAN

3 ομάδες κοιτώντας την απόσταση του 4ου γείτονα

K-means





ΣΙΣ με MAX-link

Κριτήρια Ορθότητας Συσταδοποίησης

1. Υπάρχει τάση ομαδοποίησης (clustering tendency), δηλαδή μη τυχαία δομή στο σύνολο των δεδομένων;

2. Σύγκριση των αποτελεσμάτων της ανάλυσης της ομαδοποίησης με κάποια ήδη γνωστά αποτελέσματα, πχ κάποια ετικέτα που ήδη έχει δοθεί για μια συστάδα
3. Πόσο καλά ταιριάζουν τα αποτελέσματα της ανάλυσης με τα δεδομένα χωρίς αναφορά σε εξωτερική πληροφορία, χρησιμοποιώντας μόνο τα δεδομένα

4. Σύγκριση των αποτελεσμάτων δυο διαφορετικών συσταδοποιήσεων για να αποφασιστεί ποια είναι καλύτερη.

5. Καθορισμός του «σωστού» αριθμού συστάδων

Τα 2, 3 και 4 μπορεί να αφορούν είτε την ολική συσταδοποίηση είτε τη κάθε συστάδα χωριστά

Μετρήσεις Ποιότητας Συσταδοποίησης

Οι μετρήσεις για την ποιότητα (το πόσο καλή) είναι μια συσταδοποίηση ανήκουν σε μία από τις παρακάτω τρεις κατηγορίες:

- **Με** επίβλεψη (supervised) - Εξωτερικό Ευρετήριο (External Index): Υπάρχει εξωτερική πληροφορία (πληροφορία εκτός των δεδομένων),
  πχ ετικέτες για τις συστάδες
  Μετράμε πόσο οι περιγραφές των συστάδων ταιριάζουν με τις ετικέτες των κλάσεων. – πχ Εντροπία

- **Χωρίς** επίβλεψη (unsupervised) Εσωτερικό Ευρετήριο (Internal Index):
  Εκτιμάμε το πόσο καλή είναι μια συσταδοποίηση χωρίς παροχή εξωτερικής πληροφορίας - πχ EES
  - Συνεκτικότητα (cohesion)
  - Διακριτότητα ή διαχωρισμός (separation)

Μετρήσεις Ποιότητας Συσταδοποίησης

- Συγκριτικοί -Σχετικό Ευρετήριο (Relative Index):

Χρησιμοποιείται για τη σύγκριση δυο διαφορετικών συσταδοποιήσεων ή συστάδων - Συχνά για αυτό το σκοπό χρησιμοποιείται ένα εσωτερικό ή εξωτερικό ευρετήριο

Εσωτερικό, πχ δυο k-means συσταδοποιήσεις με βάση το SSE

Κριτήρια vs Ευρετήρια – κριτήριο: η γενική στρατηγική και ευρετήριο η αριθμητική μέτρηση που υλοποιεί το κριτήριο

Χαρακτηρισμός Ποιότητας Συσταδοποίησης χωρίς Επίβλεψη

- Χρήση Συνοχής και Διαχωρισμού
- Χρήση Πίνακα Γειτνίασης

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **χωρίς** Επίβλεψη: **Συνοχή και Διαχωρισμός**

Έχουμε μια συσταδοποίηση (ένα σύνολο συστάδων): Πόσο «καλή/έγκυρη» είναι

Δύο μέτρα:

- Ένα για να χαρακτηρίσουμε **κάθε συστάδα ξεχωριστά** (cohesion – συνοχή) : πόσο κοντά (όμοια) είναι τα σημεία κάθε συστάδας
- Ένα **για τις συστάδες μεταξύ τους** (separation – διαχωρισμός): πόσο μακριά (ανόμοιες) είναι δύο συστάδες)

Συνδυασμός της συνοχής και του διαχωρισμού για το χαρακτηρισμό συνολικά της συσταδοποίησης

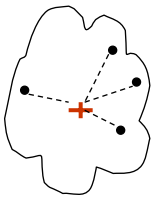
Χαρακτηρισμός Ποιότητας Συσταδοποίησης

Δύο γενικοί ορισμοί για κάθε μέτρο:

1. Prototype-based (βάσει προτύπων): με βάση το «κεντρικό σημείο» (κέντρο βάρους) κάθε συστάδας
2. Graph-based (Βάσει γραφημάτων): με βάση τις ανά-δύο αποστάσεις των σημείων

Συνοχή

Συνοχή βασισμένη σε <u>πρότυπα</u>



$$cohesion(C_i) = \sum_{x \in C_i}^{n} proximity(x, c_i)$$

Όπου: x (σημείο της συστάδας) – c_i πρότυπο της συστάδας (π.χ., διάμεσος, κέντρο βάρους)

Proximity (εγγύτητα) μπορεί να είναι η ομοιότητα ή η απόσταση των σημείων

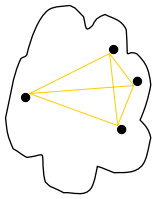
Π.χ., το άθροισμα των αποστάσεων όλων των σημείων της συστάδας από το κέντρο βάρους της συστάδας

Όπου c_i το κεντρικό σημείο (X_0) στον BIRCH

ακτίνα R στον BIRCH/k-means

Συνοχή

Συνοχή βασισμένη σε <u>γραφήματα</u>



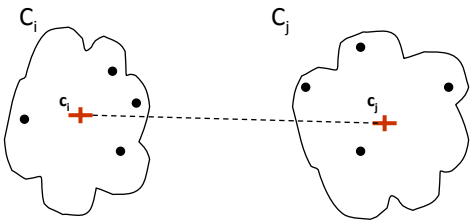
$$cohesion\,(C_i) = \sum_{\substack{x \in C_i \\ y \in C_i}}^{n} proximity(x, y)$$

Η συνοχή μιας συστάδας (cluster cohesion) είναι το άθροισμα της εγγύτητας (συνήθως απόσταση) μεταξύ <u>όλων</u> των σημείων της συστάδας.

αντιστοιχεί στο D - διάμετρο στον BIRCH

Διαχωρισμός

Διαχωρισμός βασισμένος σε <u>πρότυπα</u>



$$separation(C_i) = proximity(c_i, c)$$

Την απόσταση του πρότυπου μιας συστάδας από το σημείο c -> το συνολικό κέντρο όλων των σημείων

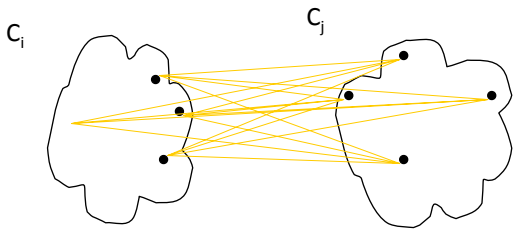
$$separation(C_i, C_j) = proximity(c_i, c_j)$$

Την απόσταση μεταξύ των προτύπων (π.χ., κέντρα βάρους) κάθε συστάδας

αντιστοιχεί στα D0 (D1) στον BIRCH

Διαχωρισμός

Διαχωρισμός βασισμένος σε <u>γραφήματα</u>



$$separation(C_i, C_j) = \sum_{\substack{x \in C_i \\ y \in C_j}}^{n} proximity(x, y)$$

Όλες τις αποστάσεις μεταξύ των σημείων της μιας συστάδας από τα σημεία της άλλης συστάδας

αντιστοιχεί στο D2 στον BIRCH

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **χωρίς** Επίβλεψη: **Συνοχή και Διαχωρισμός**

$$overall - validity = \sum_{i=1}^{k} w_i \, validity(C_i)$$

Όπου το βάρος (w_i) μπορεί να είναι πχ ανάλογο του μεγέθους της συστάδας ή η τετραγωνική ρίζα της συνεκτικότητας ή 1

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **χωρίς** Επίβλεψη: **Συνοχή και Διαχωρισμός**

Συνολική Συνοχή

$$overall - cohesion = \sum_{i=1}^{k} w_i \, \text{cohesion}(C_i)$$

Άθροισμα συνοχής κάθε συστάδας – το βάρος μπορεί να εξαρτάται πχ από το μέγεθος κάθε συστάδας

Συνολικός Διαχωρισμός

$$overall - separation = \sum_{i=1}^{k} w_i \, \text{separation}(C_i)$$

Άθροισμα διαχωρισμού των συστάδων

Συνολικός Χαρακτηρισμός Ποιότητας για τη συσταδοποίηση

$$overall - validity = \sum_{i=1}^{k} \frac{\text{seperation}(C_i)}{\text{cohesion}(C_i)}$$

Σχέση Συνοχής και Διαχωρισμού (για Ευκλείδειες αποστάσεις)

Σχέση μεταξύ συνοχής προτύπου και γραφήματος

Έστω Ευκλείδεια απόσταση, **σχέση SSE** (λάθους με βάση το άθροισμα των τετραγώνων των αποστάσεων) **με συνοχή** (πόσο στενά σχετιζόμενα είναι τα αντικείμενα μιας συστάδας);

$$\text{ανά συστάδα} - SSE = \sum_{\substack{x \in C_i \\ y \in C_i}}^{n} \text{distance}(x, y)^2$$

$$\text{ανά συστάδα} - SSE = \sum_{x \in C_i}^{n} \text{distance}(x, c_i)^2$$

Αποδεικνύεται ότι

$$SSE = \sum_{x \in C_i} dist^2(x, c_i) = 1/2m_i \sum_{x \in C_i} \sum_{y \in C_i} dist(x, y)^2$$

Όπου m ο αριθμός σημείων της συστάδας

Δηλαδή, είτε πάρουμε την απόσταση από το κέντρο (πρότυπο) είτε το μέσο όρο των ανά δύο αποστάσεων των σημείων είναι το ίδιο

Σχέση διαμέτρου και ακτίνας

Σχέση Συνοχής και Διαχωρισμού (για Ευκλείδειες αποστάσεις)

Σχέση μεταξύ διαχωρισμού προτύπου και γραφήματος

Έστω Ευκλείδεια απόσταση, **σχέση SSB (group sum of squares) με διαχωρισμό** (πόσο μακριά είναι οι συστάδες);

$$cluster - SSB = dist(c_i, c)^2$$

$$(ολικό-)SSB = \sum_{i=1}^{K} m_i \, dist(c_i, c)^2$$

Το ολικό κέντρο (σημείο c στους τύπους) είναι το σημείο που προκύπτει αν πάρουμε το μέσο (mean) των κέντρων όλων των συστάδων

Μέγεθος συστάδας

Αποδεικνύεται ότι

Ισομεγέθεις συστάδες

$$ολικό - SSB = \sum_{x \in C_i} m_i \, dist^2(c_i, c) = \frac{1}{2K} \sum_{i=1}^{K} \sum_{j=1}^{K} \frac{m}{K} dist(c_i, c_j)^2 \qquad m_i = m / K$$

Δηλαδή, είτε πάρουμε την απόσταση των κέντρων (προτύπων) κάθε συστάδας από το ολικό κέντρο (πρότυπο) είτε το μέσο όρο των ανά δύο αποστάσεων των κέντρων (προτύπων) κάθε συστάδας είναι το ίδιο

Σχέση Συνοχής και Διαχωρισμού (για Ευκλείδειες αποστάσεις)

Αποδεικνύεται ότι

Total SSB + Total SSE = σταθερά

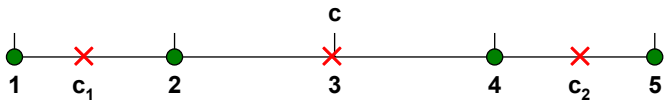
$$TSS = \sum_{i=1}^{K} \sum_{x \in C_i} (x - c)^2$$

Ίσο με το τετράγωνο των αποστάσεων όλων των σημείων από το ολικό μέσο

Ελαχιστοποίηση της SSE (συνοχής) => Μεγιστοποίηση του SSB (διαχωρισμού)

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **χωρίς** Επίβλεψη: **Συνοχή και Διαχωρισμός**

Συνολική Συνοχή + Συνολικός Διαχωρισμός = σταθερά
Total-SSE + Total-SSB = σταθερά



K = 1 cluster:

$$total - SSE = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$total - SSB = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K = 2 clusters:

$$total - SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$total - SSB = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

Ίση με το άθροισμα του τετραγώνου της απόστασης όλων των σημείων από το κέντρο βάρους τους

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **χωρίς** Επίβλεψη: **Συνοχή και Διαχωρισμός**

Μπορούν να χρησιμοποιηθούν για τη βελτίωση της συσταδοποίησης

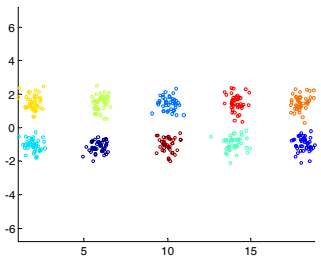
Πχ μια συστάδα με κακή συνοχή μπορεί να χρειαστεί να διασπαστεί

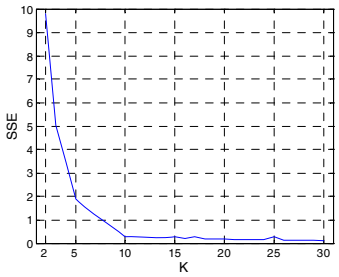
Δυο συστάδες όχι καλά διαχωρισμένες μπορεί να συγχωνευτούν

- Το πόσο καλή είναι μια συσταδοποίηση
- Το ποσό καλή είναι μια συστάδα
- Το ποσό καλό είναι ένα σημείο σε μια συστάδα

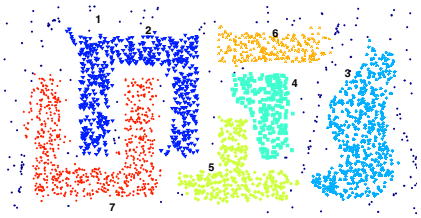
Χρήση για καθορισμό του πλήθους συστάδων

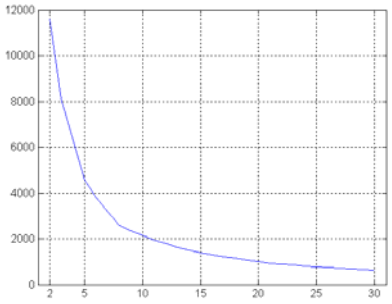
Χρήση SSE (άθροισμα τετραγώνου αποστάσεων) για υπολογισμό του σωστού αριθμού συστάδων χρησιμοποιώντας τον K-means
(K = 5 και 10 φαίνονται καλές τιμές)





Χρήση για καθορισμό του πλήθους συστάδων





Συντελεστής Σκιαγράφησης

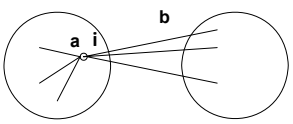
Silhouette Coefficient (συντελεστής σκιαγράφησης)

Για κάθε σημείο *i*

Υπολογισμός ***a*** = μέση απόσταση του *i* από τα σημεία της συστάδας
Υπολογισμός ***b*** = μέση απόσταση του *i* από όλα τα σημεία κάθε άλλης συστάδας – επιλογή του μικρότερου, δηλαδή μέση απόσταση από την κοντινότερη συστάδα

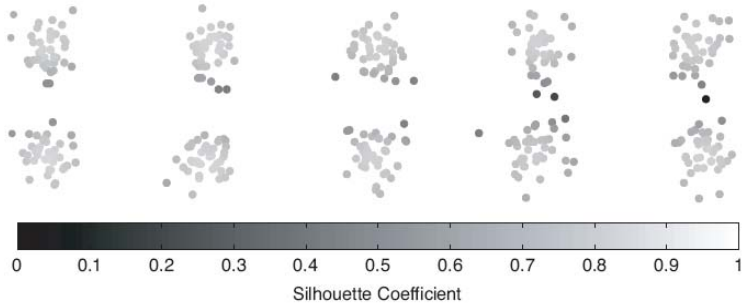
$s = 1 - a/b$ αν $a < b$, (ή $s = b/a - 1$ αν $a \geq b$, η μη συνηθισμένη περίπτωση)

Συνήθως μεταξύ του 0 και του 1
Όσο πιο κοντά στο 1, τόσο το καλύτερο



Μπορεί να χρησιμοποιηθεί και για μια συστάδα ή συσταδοποίηση θεωρώντας μέσες τιμές για όλα τα σημεία τους ή συστάδες

Συντελεστής Σκιαγράφησης



Ο συντελεστής σκιαγράφησης για σημεία στις 10 συστάδες

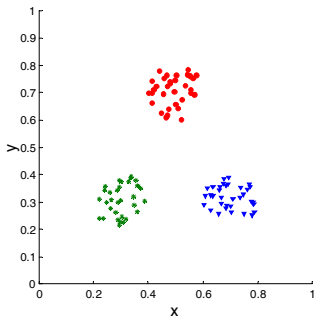
Πόσο «κεντρικό» είναι ένα σημείο για μία συστάδα (όσο πιο ανοιχτόχρωμο τόσο το καλύτερο)

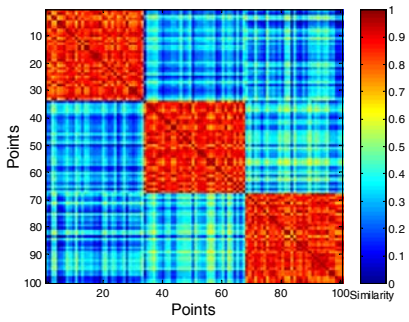
Πίνακας Εγγύτητας

Αναδιατάσσουμε τα σημεία στον πίνακα γειτνίασης ή εγγύτητα (δηλαδή, στον πίνακα με τις αποστάσεις) έτσι ώστε τα σημεία που ανήκουν στην ίδια συστάδα να είναι γειτονικά

Συγκεκριμένα, τα διατάσουμε με βάση τη συστάδα:

Σημεία Συστάδας 1, Σημεία Συστάδας 2, Σημεία Συστάδας 3

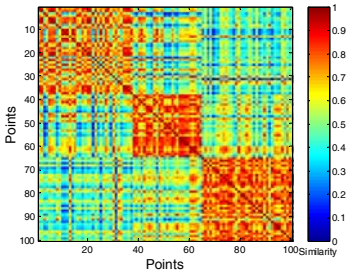


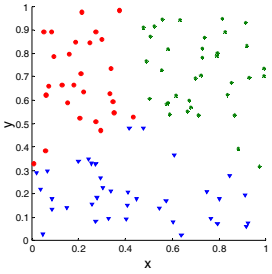


Σημείωση: similarity = 1 – (d –min_d)/(max_d – min_d)

ΔΙΑΓΩΝΙΟΣ ΜΠΛΟΚ ΠΙΝΑΚΑΣ

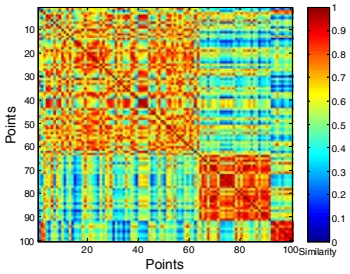
Πίνακας Εγγύτητας

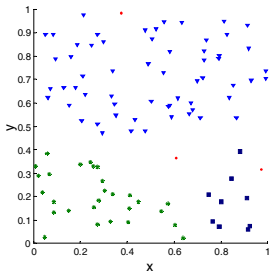




K-means

Πίνακας Εγγύτητας

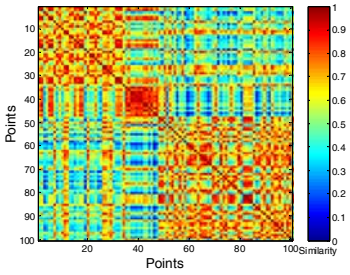


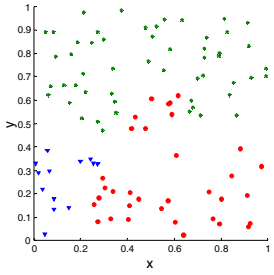


DBSCAN

Κάποιες συστάδες ακόμα και σε τυχαία δεδομένα

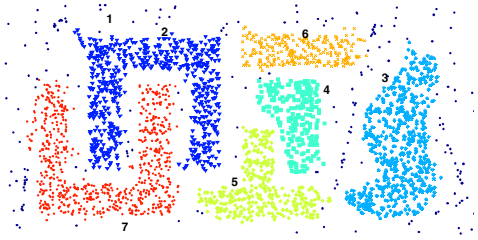
Πίνακας Εγγύτητας

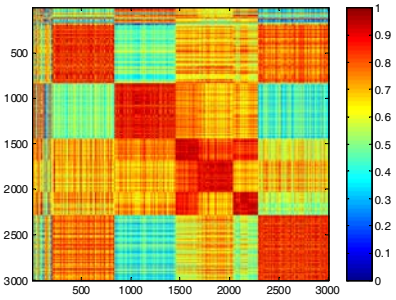




ΣΙΣ-max

Πίνακας Εγγύτητας





DBSCAN

Πίνακας Εγγύτητας

Ειδικά για ιεραρχικούς αλγόριθμους

Cophenetic distance (συμφαινετική απόσταση): είναι η απόσταση (proximity) όταν ο αλγόριθμος τοποθετεί τα δυο σημεία στην ίδια συστάδα για πρώτη φορά

Πχ συγχωνεύω τα σημεία του C1 με τα σημεία του C2 σε απόσταση 0.1, όλα τα σημεία του C1 απέχουν από το C2 0.1

Χαρακτηρισμός Ποιότητας Συσταδοποίησης **με** Επίβλεψη:

Επίσης υπάρχουν

- μέτρα με επίβλεψη
- μέτρα βασισμένα σε συσχέτιση (correlation)

που θα δούμε αφού μιλήσουμε για κατηγοριοποίηση

Πίνακας Εγγύτητας (συσχέτιση)

Δύο Πίνακες

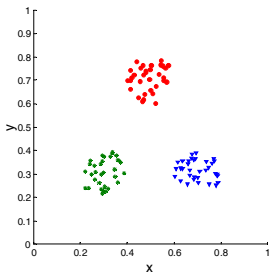
Πίνακας Εγγύτητας (proximity matrix)
ο πίνακας με την ομοιότητα των σημείων

Πίνακας Εμφάνισης ("incidence" matrix)
Μια γραμμή και μια στήλη για κάθε σημείο
Μια εγγραφή είναι **1** αν το αντίστοιχο ζευγάρι σημείων ανήκει στην ίδια συστάδα
Μια εγγραφή είναι **0** αν το αντίστοιχο ζευγάρι σημείων ανήκει σε διαφορετική συστάδα

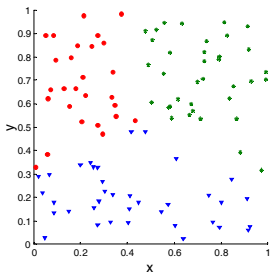
Υπολογισμός της **συσχέτισης (correlation)** των δύο πινάκων

Πίνακας Εγγύτητας (συσχέτιση)

Υπολογισμός correlation των δύο πινάκων όταν χρησιμοποιείται ο K-means στα παρακάτω σύνολα



Corr = -0.9235



Corr = -0.5810

Πίνακας Εγγύτητας (συσχέτιση)

Υψηλή συσχέτιση σημαίνει ότι τα σημεία που ανήκουν στην ίδια συστάδα είναι κοντινά μεταξύ τους

- Δεν είναι καλή μέτρηση για κάποιες συστάδες που βασίζονται σε πυκνότητα και σε συνέχεια (contiguity)

- Επειδή, οι δυο πίνακες είναι συμμετρικοί, χρειάζεται ο υπολογισμός n(n-1) / 2 εγγραφών

BIRCH

T. Zhang, R. Ramakrishnan and M. Linvy. BIRCH: An Efficient Data Clustering Method for Very Large Databases, SIGMOD 1996

BIRCH

Μεγάλα Σύνολα Δεδομένων

Περιορισμένη μνήμη (πολύ μικρότερη από το μέγεθος των δεδομένων)

ΣΤΟΧΟΣ: μείωση του χρόνου εισόδου/εξόδου (I/O)

- Κόστος I/O *γραμμικό* στο μέγεθος του συνόλου δεδομένων
  - Αρκεί ένα πέρασμα (scan) των δεδομένων
  - Ένα ή περισσότερα επιπρόσθετα περάσματα για βελτίωση της ποιότητας της συσταδοποίησης

BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies

BIRCH

Δύο βασικά χαρακτηριστικά:

- Ιεραρχική συσταδοποίηση, κρατάμε το δενδρόγραμμα σε ένα δένδρο (ιεραρχία)
- Αντί να κρατάμε όλα τα σημεία μιας συστάδας κρατάμε κάποια «στατιστικά» για κάθε συστάδα και για τις σχέσεις μεταξύ των συστάδων

Ποια είναι αυτά τα στοιχεία;

Για μια συστάδα N σημείων: $\{\vec{X}_i\}$

Centroid (κέντρο βάρους): $\vec{X0} = \frac{\sum_{i=1}^{N} \vec{X}_i}{N}$

Radius (ακτίνα): μέση απόσταση των σημείων της συστάδας από το κέντρο βάρους

$$R = \left(\frac{\sum_{i=1}^{N} (\vec{X}_i - \vec{X0})^2}{N}\right)^{\frac{1}{2}}$$

Συνοχή βασισμένη σε κεντρικά σημεία

Diameter (διάμετρος): μέση ανά-δύο απόσταση των σημείων της συστάδας

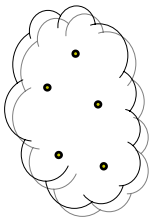
$$D = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (\vec{X}_i - \vec{X}_j)^2}{N(N-1)}\right)^{\frac{1}{2}}$$

Συνοχή βασισμένη σε γραφήματα

Έστω δυο συστάδες (στατιστικά στοιχεία για το διαχωρισμό)

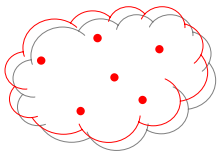
Συστάδα $\{X_i\}$:
$i = 1, 2, \ldots, N_1$

X_i



Συστάδα $\{X_j\}$:
$j = N_1+1, N_1+2, \ldots, N_1+N_2$

X_j



Μεταξύ δυο συστάδων

Η απόσταση μεταξύ των κέντρων βάρους των δυο συστάδων

Διαχωρισμός βασισμένος σε κεντρικά σημεία

centroid Euclidean distance

$$D0 = ((\vec{X}0_1 - \vec{X}0_2)^2)^{\frac{1}{2}}$$

centroid Manhattan distance

$$D1 = |\vec{X}0_1 - \vec{X}0_2| = \sum_{i=1}^{d} |\vec{X}0_1^{(i)} - \vec{X}0_1^{(i)}|$$

average inter-cluster (D2) μέση απόσταση των σημείων της μιας συστάδας από τα σημεία της άλλης

Διαχωρισμός με βάση γραφήματα

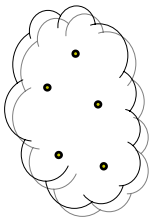
$$D2 = \left(\frac{\sum_{i=1}^{N_1}\sum_{j=N_1+1}^{N_1+N_2}(\vec{X}_i - \vec{X}_j)^2}{N_1 N_2}\right)^{\frac{1}{2}}$$

Συγχώνευση Συστάδων

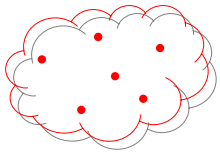
Συστάδα $\{X_i\}$:
$i = 1, 2, \ldots, N_1$

Συστάδα $\{X_j\}$:
$j = N_1+1, N_1+2, \ldots, N_1+N_2$

X_i



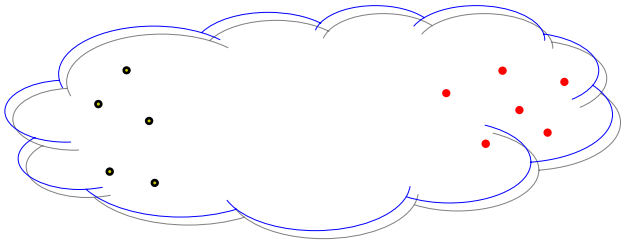
X_j



BIRCH

Συστάδα $X_k = \{X_i\} + \{X_j\}$:

I = 1, 2, …, N_1, N_1+1, N_1+2, …, N_1+N_2



Η νέα συστάδα έχει το σύνολο των σημείων των δύο συστάδων

BIRCH

intra-cluster (D3) μέση απόσταση όλων των σημείων

$$D3 = \left(\frac{\sum_{i=1}^{N_1+N_2}\sum_{j=1}^{N_1+N_2}(\vec{X}_i - \vec{X}_j)^2}{(N_1+N_2)(N_1+N_2-1)}\right)^{\frac{1}{2}}$$

Είναι ουσιαστικά η διάμετρος D της συγχωνευμένης συστάδας

variance increase (D4)

Νέα Απόσταση

$$D4 = \left(\sum_{k=1}^{N_1+N_2}(\vec{X}_k - \frac{\sum_{l=1}^{N_1+N_2}\vec{X}_l}{N_1+N_2})^2 - \sum_{i=1}^{N_1}(\vec{X}_i - \frac{\sum_{l=1}^{N_1}\vec{X}_l}{N_1})^2 - \sum_{j=N_1+1}^{N_1+N_2}(\vec{X}_j - \frac{\sum_{l=N_1+1}^{N_1+N_2}\vec{X}_l}{N_2})^2\right)^{\frac{1}{2}}$$

Απόσταση στο C_i Απόσταση στο C_j

BIRCH: CF

Clustering Feature (CF): μια περίληψη μιας συστάδας δεδομένων

$$\mathbf{CF} = (N, \vec{LS}, SS)$$

Μια τριάδα
N: αριθμός-σημείων,
LS: γραμμικό-άθροισμα-σημείων-συστάδας,
SS: άθροισμα-τετραγώνου-σημείων-συστάδας)

Δοθείσας μια συστάδας $\{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N\}$

$$\vec{LS} = \sum_{i=1}^{N} \vec{X}_i$$
$$SS = \sum_{i=1}^{N} \vec{X}_i^{\,2}$$

BIRCH: CF

Σημαντική (προσθετική) ιδιότητα:

Για το CF της νέας συστάδας, ισχύει:

$$\mathbf{CF_1} + \mathbf{CF_2} = (N_1 + N_2, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2)$$

BIRCH: CF

- CF εγγραφές είναι συνοπτικές – πολύ λιγότερη πληροφορία από ότι όλα τα σημεία μιας συστάδας

- Λόγω της προσθετικής ιδιότητας μπορούμε να συγχωνεύσουμε δυο υπο-συστάδες σταδιακά

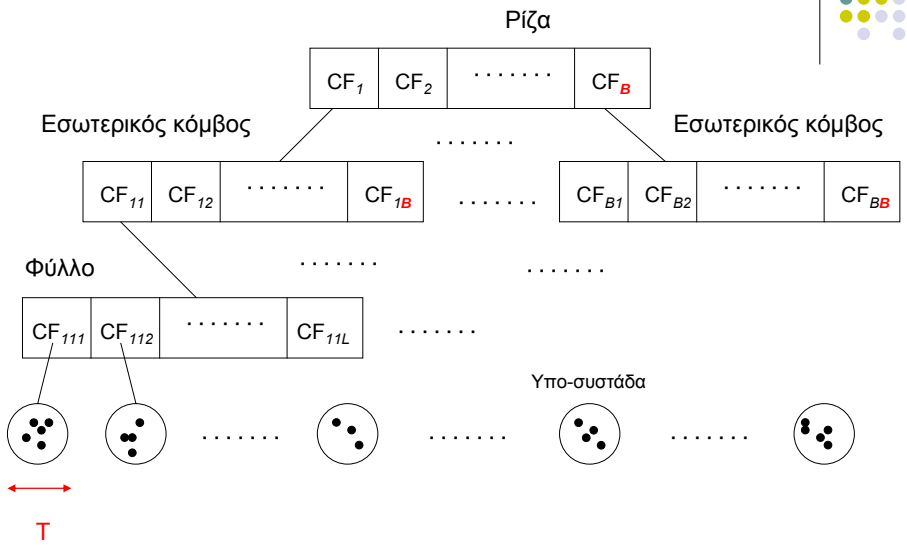
Μια εγγραφή CF έχει αρκετή πληροφορία για να υπολογίσουμε τα D0-D4

BIRCH

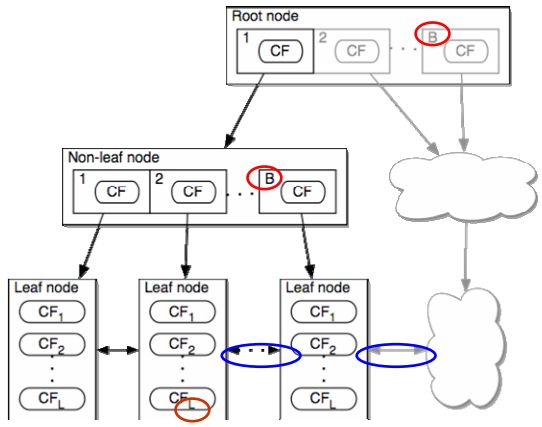
Ιεραρχικός αλγόριθμος

Χτίζει σταδιακά καθώς διαβάζει τα δεδομένα ένα δεντρό-γραμμα του οποίου κόμβοι είναι οι τιμές CF που περιγράφουν τα δεδομένα κάθε υπο-συστάδας

BIRCH: CF-δέντρο

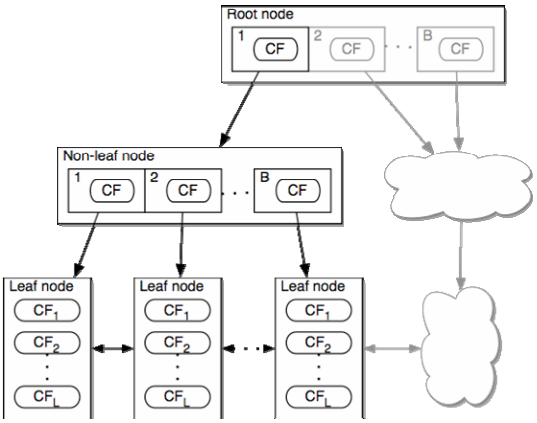


BIRCH: CF-δέντρο



- Κάθε εσωτερικός κόμβος μια υπό-συστάδα που αποτελείται από τις υπό-συστάδες των παιδιών του
- Κάθε *εσωτερικός κόμβος* περιέχει έναν αριθμό **B** από παιδιά, δηλαδή, υπό-συστάδες (παράγοντας διακλάδωσης) εγγραφές <CF_i, παιδί$_i$>
- Κάθε *φύλλο* περιέχει έναν αριθμό **L** από υπό-συστάδες (δηλαδή, το πολύ **L** CF εγγραφές [CFi] και <prev>, <next> pointers)

BIRCH: CF-δέντρο



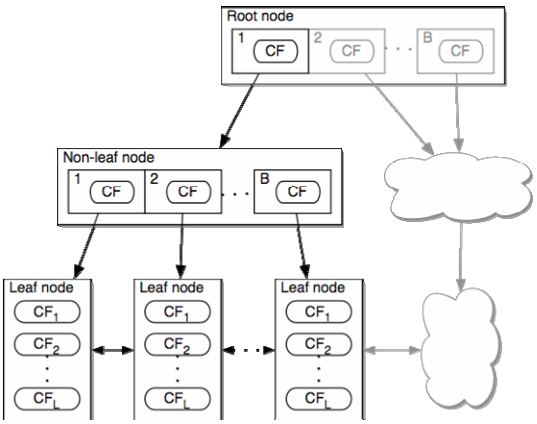
B, L

- Όπως σε όλες τις σχετικές δομές απαιτούμε κάθε κόμβος του δέντρου να χωρά σε μια σελίδα (block)

Το μέγεθος των κόμβων (B, L) καθορίζεται από τη διάσταση των δεδομένων (πόσο μεγάλο είναι το CF) και το μέγεθος της σελίδας (που δίνεται ως είσοδος)

BIRCH: CF-δέντρο

Κατώφλι ομοιότητας



Κάθε υποσυστάδα ενός φύλλου πρέπει να έχει

Διάμετρο (ή ακτίνα) μικρότερη από κάποιο κατώφλι **T**

(μέγιστη απόσταση των σημείων κάθε συστάδας)

Το μέγεθος του T καθορίζει το μέγεθος του δέντρου

Όσο πιο μεγάλο είναι το T, τόσο μικρότερο είναι το δέντρο

BIRCH: CF δέντρο

Συνοπτικά, το CF-δέντρο είναι ένα ισοζυγισμένο δέντρο με δυο παραμέτρους

- Παράγοντα διακλάδωσης **B** (που καθορίζεται από το μέγεθος της σελίδας)
- Κατώφλι **T** (που καθορίζει την *ποιότητα* της συσταδοποίησης)

BIRCH: CF-δέντρο

Για ένα φύλλο:

$$LS = \sum_{P_i \in N} \vec{P}_i$$

$$SS = \sum_{P_i \in N} \left|\vec{P}_i\right|^2$$

Για κάθε εσωτερικό κόμβο που έχει παιδιά τα N_1, N_2, …, N_k

$$\overrightarrow{LS} = \sum_{i=1}^{k} \overrightarrow{LS} \text{ of } N_i$$

$$SS = \sum_{i=1}^{k} SS \text{ of } N_i$$

BIRCH: CF-δέντρο εισαγωγή στοιχείου

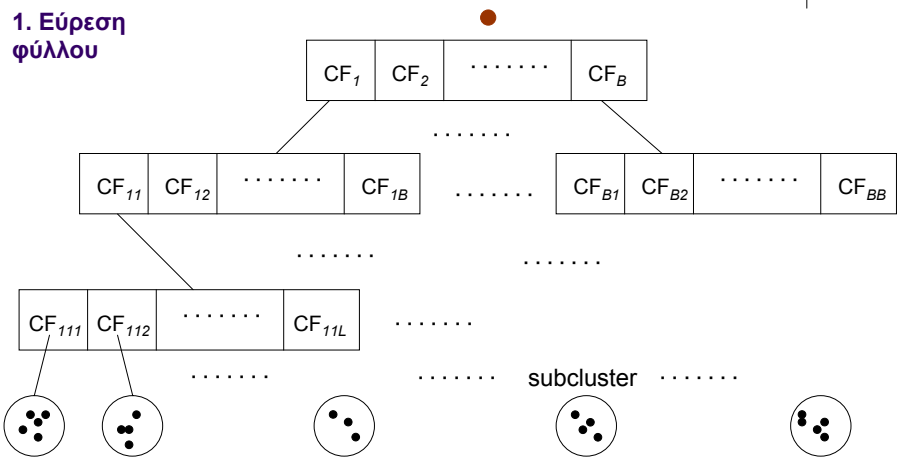
- Ο αλγόριθμος διαβάζει (scan) τα δεδομένα και τα εισάγει στο CF δέντρο ένα-ένα
- Η εισαγωγή ενός στοιχείου στο CF-δέντρο γίνεται με top-down διάσχιση ξεκινώντας από τη ρίζα με βάση μια συνάρτηση απόστασης Distance(σημείο, cluster)
  - Χρήση της D0, D1, D2, D3 ή D4
- Κάθε σημείο εισάγεται στην κοντινότερη υπό-συστάδα που υπάρχει σε κάποιο από τα φύλλα

BIRCH: CF-δέντρο εισαγωγή στοιχείου

1. Εύρεση κατάλληλου φύλου
   αν το φύλλο μπορεί να το απορροφήσει
   (διάμετρος παραμένει <= T) ok,
   Αλλιώς 3
2. Ενημέρωση του φύλλου
3. Διάσπαση φύλλου
4. Ενημέρωση τιμής CF

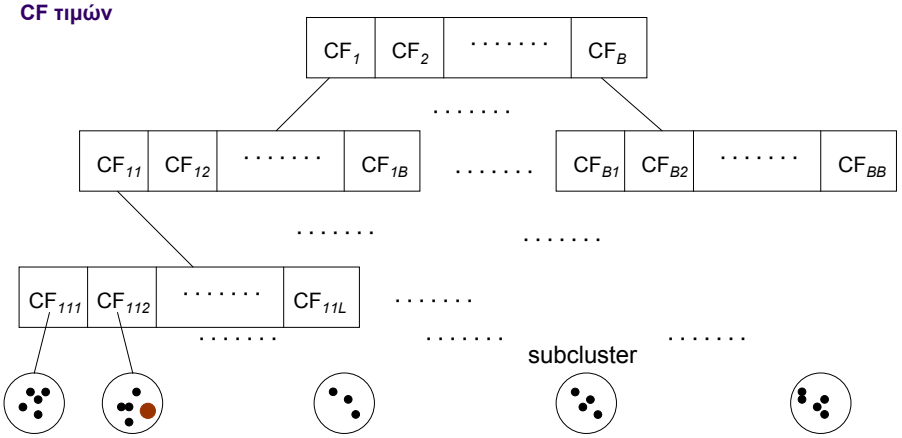
BIRCH: CF-δέντρο εισαγωγή στοιχείου

1. Εύρεση φύλλου



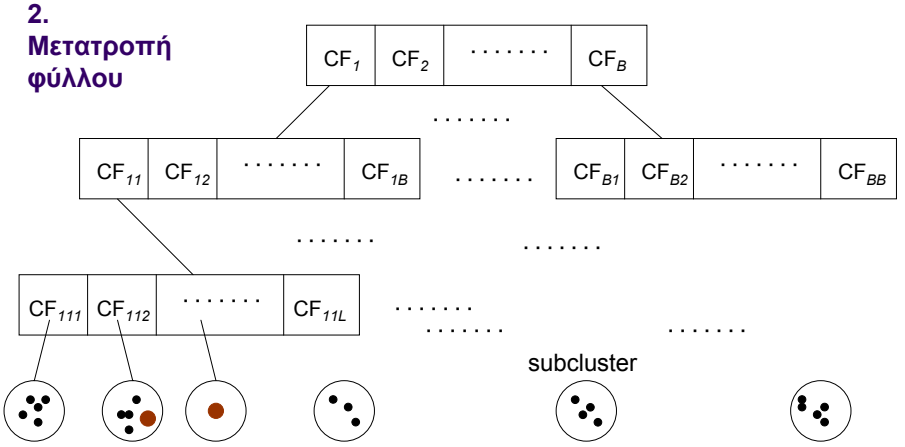
BIRCH: CF-δέντρο εισαγωγή στοιχείου

4. Τροποποίηση CF τιμών



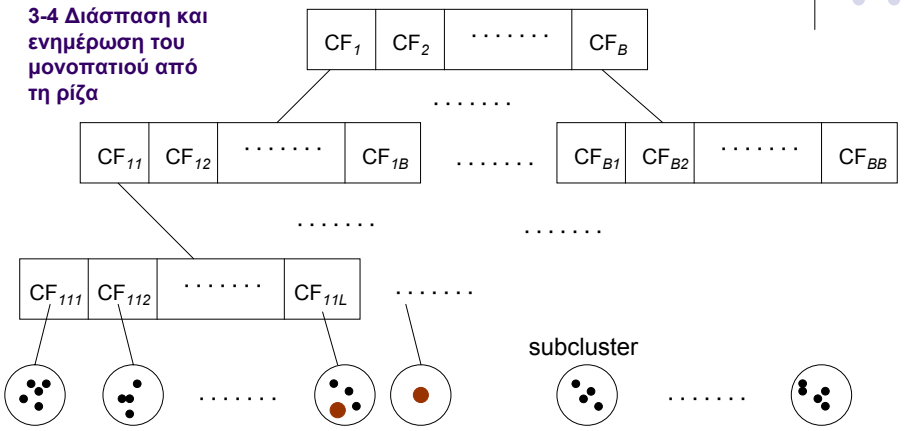
BIRCH: CF-δέντρο εισαγωγή στοιχείου

2. Μετατροπή φύλλου



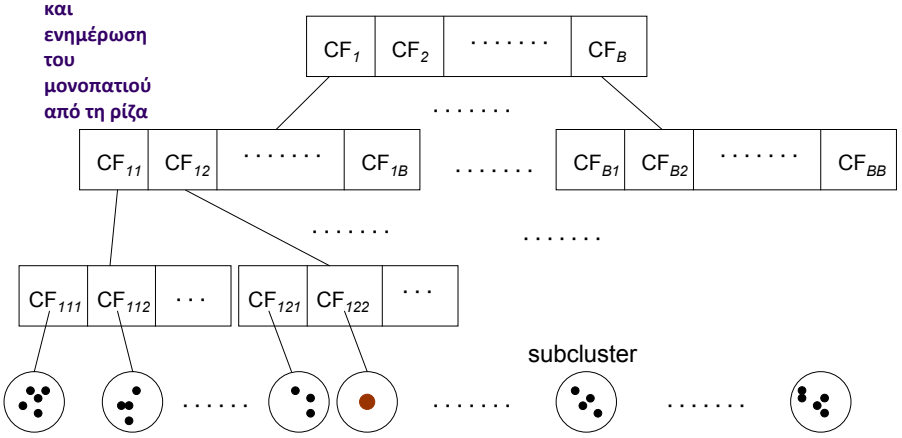
BIRCH: CF-δέντρο εισαγωγή στοιχείου

3-4 Διάσπαση και ενημέρωση του μονοπατιού από τη ρίζα



BIRCH: CF-δέντρο εισαγωγή στοιχείου

3-4 Διάσπαση και ενημέρωση του μονοπατιού από τη ρίζα



BIRCH: CF-δέντρο

- Κάθε σημείο εισάγεται στο κοντινότερη υπό-συστάδα που υπάρχει σε κάποιο από τα φύλλα
  - Αν η εισαγωγή ενός σημείου **μεγαλώσει τη διάμετρο της υποσυστάδας πάνω από T**, τότε έχουμε **δημιουργία νέας υποσυστάδας**
    - Αν η νέα συστάδα χωρά στο φύλλο, ok -> ενημέρωση προγόνων
  - Αν η νέα συστάδα δε χωρά -> υπερχείλιση στο φύλλο

BIRCH: CF-δέντρο

- Διάσπαση φύλλου (split)
  - Δημιουργία νέου φύλλου και μοίρασμα των συστάδων, πως;

Εύρεση των δύο υπό-συστάδων του φύλλου πού έχουν τη μεγαλύτερη απόσταση μεταξύ τους, έστω Ci και Cj

Αυτές οι δύο αποτελούν το κριτήριο διάσπασης των υπο-συστάδων του φύλλου – κάθε μια από αυτές σε ένα από τα δύο νέα φύλλα

όλες οι άλλες υπό-συστάδες C ανατίθενται στο φύλλο της Ci ή στο φύλλο της Cj με βάση ποια από τις δύο είναι πιο όμοια της

BIRCH: CF-δέντρο

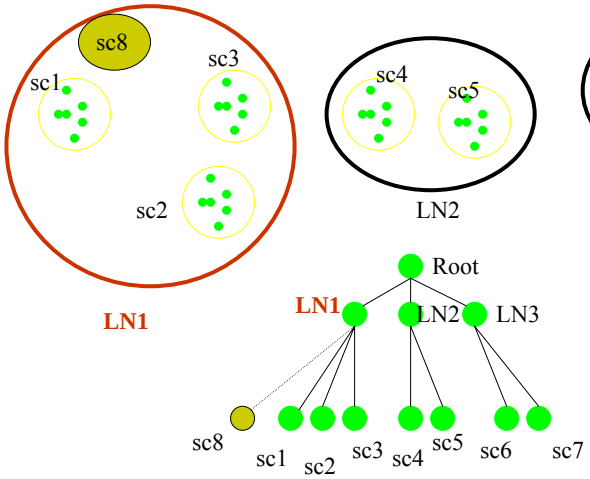
Διάσπαση φύλλου μπορεί να οδηγήσει σε υπερχείλιση εσωτερικού κόμβου (όταν περιέχει περισσότερα παιδιά από ότι ο παράγοντας διακλάδωσης)

Διάσπαση εσωτερικού κόμβου

- Οι εσωτερικοί κόμβοι διασπούνται αναδρομικά με βάση μια μέτρηση της απόσταση των συστάδων τους
- Διάσπαση της ρίζας, οδηγεί σε αύξηση του ύψους του δέντρου κατά 1

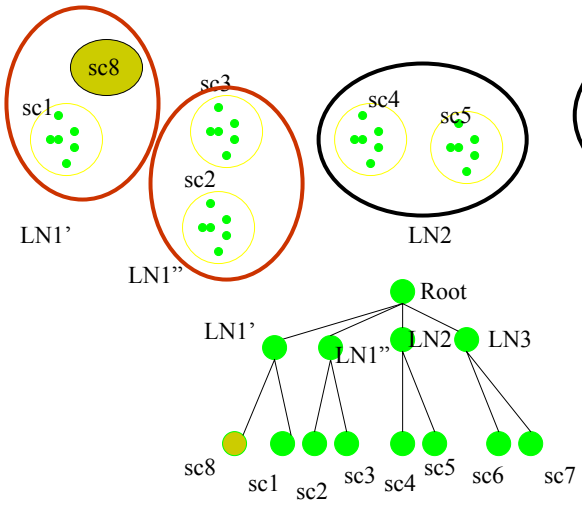
BIRCH: CF-δέντρο

Νέα υπο-συστάδα



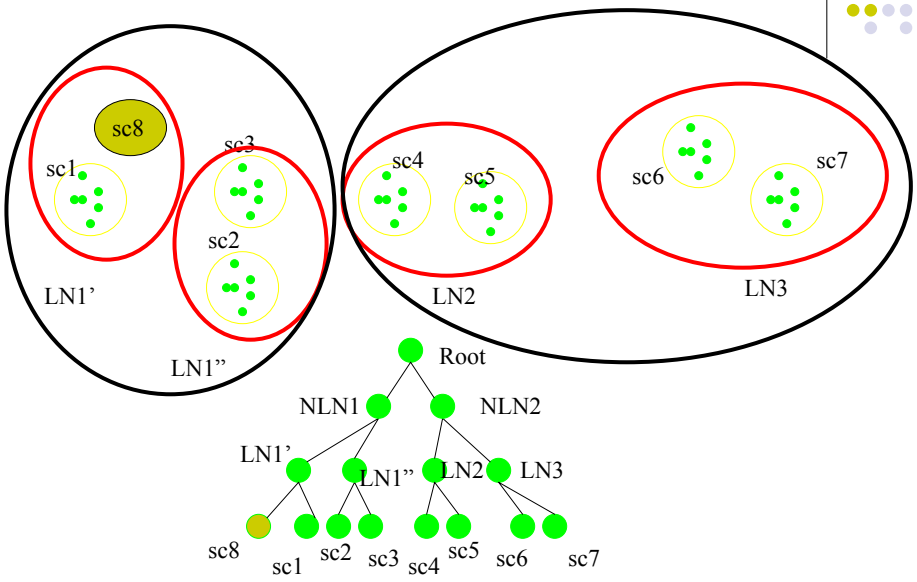
Αν ο παράγοντας διακλάδωσης του φύλλου είναι 3 => διάσπαση του LN1

BIRCH: CF-δέντρο



Αν ο παράγοντας διακλάδωσης εσωτερικού κόμβου είναι 3 => διάσπαση της ρίζας και αύξηση του ύψους

BIRCH: CF-δέντρο



BIRCH: CF-δέντρο

Οι διασπάσεις οφείλονται στο ότι ξεπερνιέται το όριο της σελίδας – μπορούν να οδηγήσουν σε κακές διασπάσεις!

Μια μικρή διόρθωση:
Όταν η διάσπαση κάποιων κόμβων τελειώνει (χωρούν σε ένα κόμβο) έστω στον κόμβο Nj κοιτάμε τον κόμβο Nj και **προσπαθούμε να συγχωνεύσουμε** τις δύο πιο κοντινές συστάδες – αν αυτές δε προέκυψαν από την πιο πρόσφατη διάσπαση

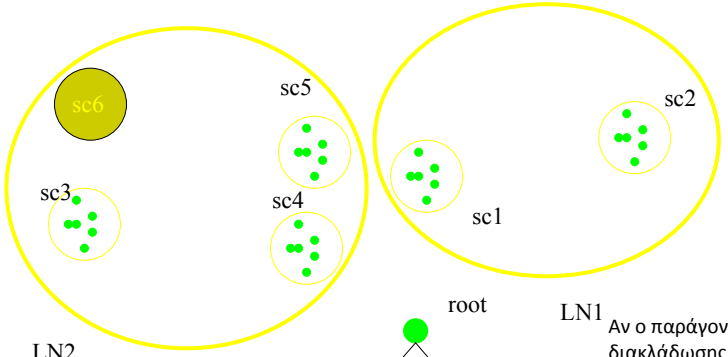
Αυτό σημαίνει ότι πρέπει να συγχωνεύσουμε και τα αντίστοιχα 2 παιδιά

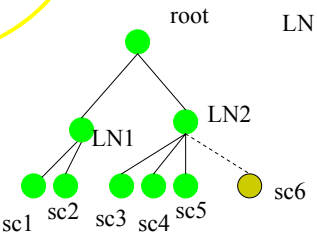
Αν χωρούν σε μια σελίδα -> ελάττωση χώρου,
Αλλιώς ανακατανέμουμε τις εγγραφές – Πως; κάνουμε πάλι διάσπαση

Τελικά ή συγχώνευση και ελευθέρωση χώρου ή καλύτερη ανακατανομή των εγγραφών σε κάποιο από τα παιδιά

BIRCH: CF-δέντρο

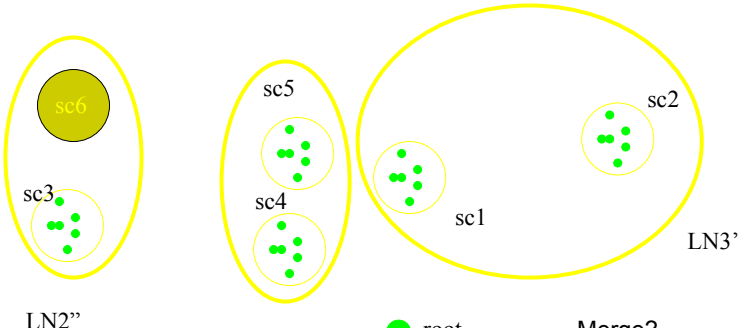
Έστω ότι η αρίθμηση των υποσυστάδων αντιστοιχεί στη σειρά δημιουργίας τους

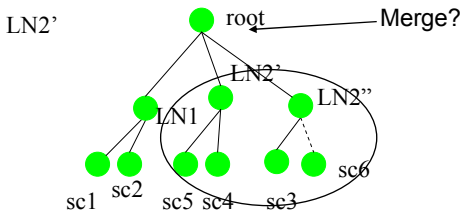




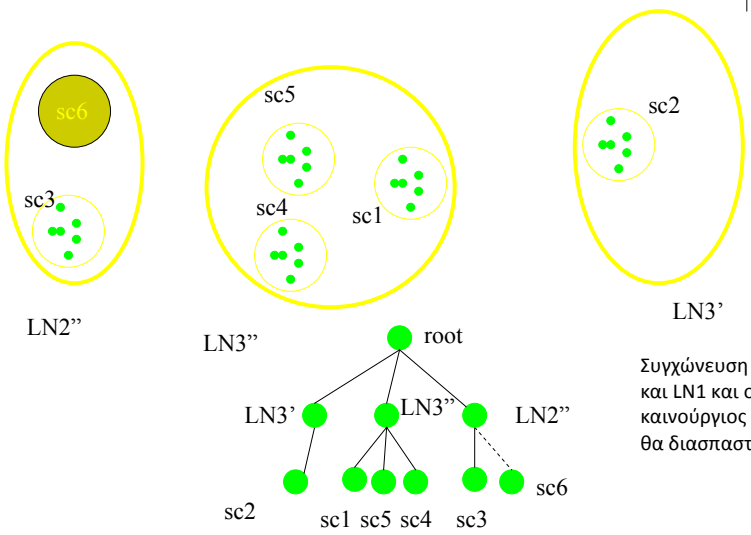
Αν ο παράγοντας διακλάδωσης του φύλλου είναι 3 => διάσπαση του LN2

BIRCH: CF-δέντρο





BIRCH: CF-δέντρο



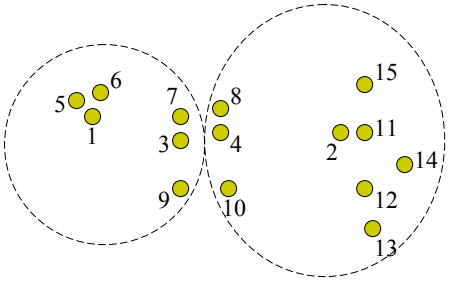
Συγχώνευση LN2’ και LN1 και ο καινούργιος κόμβος θα διασπαστεί πάλι

BIRCH: αλγόριθμος

Επειδή η κατασκευή επηρεάζεται από το μέγεθος της σελίδας:

- Οι συστάδες που δημιουργούνται μπορεί να μην είναι πραγματικές
- ανάλογα με το skew (κατανομή) και τη σειρά που έρχονται τα δεδομένα

Επίσης, αν ξανά-εισάγουμε ένα σημείο μπορεί να εισαχθεί σε διαφορετική συστάδα



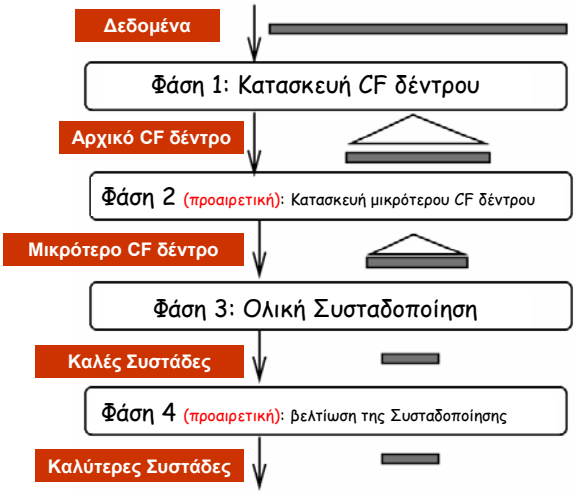
Αριθμός αντιστοιχεί στη σειρά εισαγωγής,

Έστω dist(1, 2) > T

BIRCH: αλγόριθμος

Αυτό αντιμετωπίζεται με προαιρετικές επιπρόσθετες φάσεις

BIRCH-αλγόριθμος



Φάση 1: Μια δομή κύριας μνήμης που συνοψίζει τα δεδομένα

Φάση 2: Κοιτά τα φύλλα και προσπαθεί να διώξει τους outliers και να ενοποιήσει «όμοιες» συστάδες που αντιστοιχούν σε περιοχές με πολλά σημεία

Χρειάζεται για να βελτιώσει τη Φάση 3

BIRCH

Φάση 3

Ξανα-συσταδοποιεί τα φύλλα του δέντρου

Γιατί;

Πχ κοντινές συστάδες που (έτυχε να) είναι σε διαφορετικά φύλλα

Πως;

- Για κάθε συστάδα που εμφανίζεται στα φύλλα , υπολογίζουμε το κεντρικό της σημείο (centroid) και τα θεωρούμε ως αρχικά σημεία – αυτά τα αρχικά σημεία μπορούμε να τα συσταδοποιήσουμε χρησιμοποιώντας έναν οποιαδήποτε αλγόριθμο συσταδοποίησης
  - Μπορούμε αντί ένα σημείο ανά συστάδα, κάθε συστάδα τόσες φορές όσες τα σημεία της
- Εναλλακτικά, μπορούμε να συσταδοποιήσουμε τις συστάδες ως έχουν – πχ με έναν ιεραρχικό συγκεντρωτικό αλγόριθμο

BIRCH

Φάση 4 (προαιρετική)

Χρησιμοποιεί τα κεντρικά σημεία των συστάδων που παράγει η Φάση 3 ως seeds, και αναδιανέμει όλα τα στοιχεία εισόδου (δεύτερο πέρασμα!)

Μπορεί να έχουμε και παραπάνω από ένα επιπρόσθετα περάσματα (έχει αποδειχτεί σύγκλιση)

- Εξασφαλίζει ότι όλα τα αντίγραφα ενός σημείου πάνε στην ίδια συστάδα
- Μπορούμε επίσης να βάλουμε ως ετικέτα σε κάθε σημείο, τη συστάδα που ανήκει
- Μπορούμε να απαλλαγούμε από outliers (πχ σημεία πολύ μακριά από όλα τα seeds)

Λίγα ακόμα για τη Φάση 1

Ξεκίνα με *κάποια αρχική τιμή* για το threshold (T)

Διαβάζει τα δεδομένα και τα εισάγει στο δέντρο

Αν ξεπεράσει το διαθέσιμο χώρο πριν διαβάσει όλα τα δεδομένα:
Αύξηση του threshold
Κτίσιμο νέου (μικρότερου) δέντρου ξανά-εισάγοντας τις τιμές από το παλιό δέντρο

Μόλις εισαχθούν όλες οι τιμές από το παλιό στο νέο δέντρο, Συνεχίζεται η ανάγνωση των δεδομένων από εκεί που είχε σταματήσει

Πως γίνεται η ανα-κατασκευή

Μονοπάτι-Μονοπάτι

Ανακατασκευάζουμε κάθε μονοπάτι από τη ρίζα στο φύλλο, ξεκινώντας από το πιο αριστερό μονοπάτι (old-current path)

Δημιουργούμε το new-current path

Κάθε φύλλο είτε στο new είτε στο newclosest

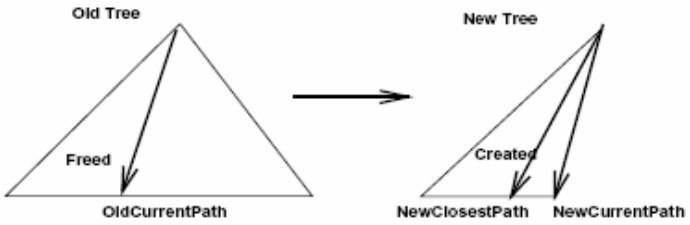


Figure 3: *Rebuilding CF Tree*

BIRCH-αλγόριθμος

1. Create the corresponding “NewCurrentPath” in the new tree
2. Insert leaf entries in “OldCurrentPath” to the new tree
   ① NewClosestPath
   ② NewCurrentPath
3. Free space in “OldCurrentPath” and “NewCurrentPath”
4. Set “OldCurrentPath” to the next path if there exists one