Πανεπιστήμιο Πειραιώς Τμήμα Πληροφορικής

Εξόρυξη Γνώσης από Δεδομένα (Data Mining)

Συσταδοποίηση
(clustering)

Γιάννης Θεοδωρίδης, Νίκος Πελέκης

Ομάδα Διαχείρισης Δεδομένων
Εργαστήριο Πληροφοριακών Συστημάτων

http://isl.cs.unipi.gr/db

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - Διαμεριστικοί Αλγόριθμοι
  - Γενετικοί Αλγόριθμοι
  - Συσταδοποίηση Μεγάλων ΒΔ

Βασισμένες κατά κύριο λόγο (αλλά όχι αποκλειστικά)
στις διαφάνειες που συνοδεύουν το βιβλίο
M. H. Dunham: Data Mining, Introductory and Advanced Topics
© Prentice Hall, 2002

Επιμέλεια Ελληνικής έκδοσης © Βασίλης Βερύκιος & Γιάννης Θεοδωρίδης, 2004-05

2

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - Διαμεριστικοί Αλγόριθμοι
  - Γενετικοί Αλγόριθμοι
  - Συσταδοποίηση Μεγάλων ΒΔ

3

Εφαρμογές Συσταδοποίησης

- **Διαμέριση** μίας ΒΔ πελατών με βάση παρόμοια πρότυπα αγοράς προϊόντων.
- Ομαδοποίηση των σπιτιών μίας πόλης σε γειτονιές με βάση παρόμοιες ιδιότητες.
- Αναγνώριση νέων ειδών φυτών
- Αναγνώριση παρόμοιων προτύπων στη χρήση του Web.
- Συσταδοποίηση vs. Κατηγοριοποίηση
  - Καμία εκ των προτέρων γνώση (αριθμός συστάδων, σημασία των συστάδων)
  - Μη Εποπτευόμενη Μάθηση

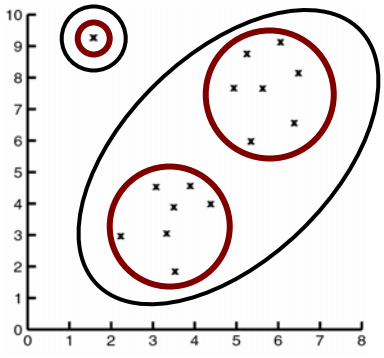
4

Ζητήματα στη Συσταδοποίηση

- Χειρισμός Ακραίων Σημείων (**outliers**)
- Δυναμικά Μεταβαλλόμενα Δεδομένα
  - αλλαγή συστάδων στην πορεία του χρόνου
- Ερμηνεία και Αξιολόγηση των αποτελεσμάτων
- Πλήθος Συστάδων
  - ο βέλτιστος αριθμός συστάδων δεν είναι εκ των προτέρων γνωστός
- Ποια δεδομένα θα χρησιμοποιηθούν
- Κλιμάκωση

5

Επίδραση Ακραίων Σημείων



- Έχουν αναπτυχθεί ειδικές τεχνικές για **ανίχνευση** / **εξόρυξη ακραίων σημείων** (outlier detection / mining)

6

Το Πρόβλημα της Συσταδοποίησης

Δοθέντων:

- μιας ΒΔ $D=\{t_1, t_2, \ldots, t_n\}$ από εγγραφές,
- ενός μέτρου ομοιότητας $sim(t_i, t_j)$ μεταξύ δύο εγγραφών της ΒΔ και
- μιας ακέραιας τιμής k,

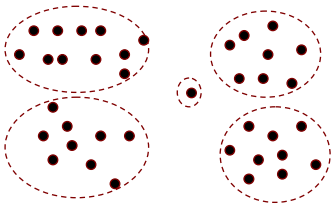
το **Πρόβλημα της Συσταδοποίησης** είναι η εύρεση μίας αντιστοίχισης $f : D \rightarrow \{1, \ldots, k\}$ όπου κάθε εγγραφή t_i της ΒΔ αντιστοιχίζεται σε μία συστάδα K_j, $1 \leq j \leq k$, έτσι ώστε:

- για κάθε εγγραφή η **ομοιότητα** μεταξύ αυτής και οποιασδήποτε εγγραφής από την ίδια συστάδα να είναι μεγαλύτερη από την ομοιότητα μεταξύ αυτής και οποιασδήποτε εγγραφής από άλλες συστάδες.

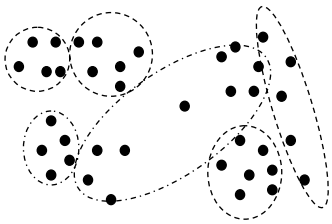
- Μία **Συστάδα**, K_j, περιέχει ακριβώς εκείνες τις πλειάδες που αντιστοιχίζονται σε αυτήν.

7

Συσταδοποίηση Σπιτιών



με βάση τη (γεωγραφική) απόσταση



με βάση κάποιο άλλο χαρακτηριστικό (π.χ. μέγεθος)

8

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - Διαμεριστικοί Αλγόριθμοι
  - Γενετικοί Αλγόριθμοι
  - Συσταδοποίηση Μεγάλων ΒΔ

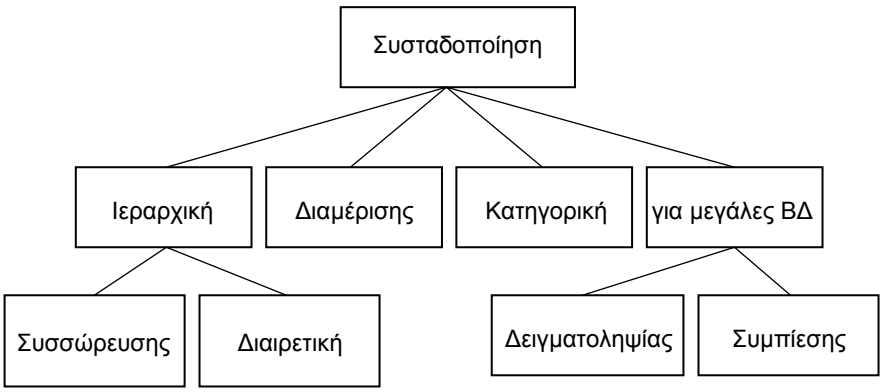
9

Τύποι Συσταδοποίησης

- **Ιεραρχική** vs. **Διαμέρισης**
  - δημιουργούνται εμφωλιασμένα σύνολα συστάδων
    - από 1 σε 2, 3, … N συστάδες (διαιρετικοί αλγόριθμοι) ή
    - από N σε N-1, ..., 2, 1 συστάδες (συσσωρευτικοί αλγόριθμοι)
  - ή δημιουργείται απευθείας ένα σύνολο k συστάδων.
- **Αυξητική** (ή **Σειριακή**) vs. **Ταυτόχρονη**
  - χειρισμός ενός στοιχείου την φορά ή όλων των στοιχείων μαζί.
- **Επικαλυπτόμενη** vs. **μη-Επικαλυπτόμενη**
  - επιτρέπεται ή όχι η τοποθέτηση ενός στοιχείου σε περισσότερες από μία συστάδες.
- Για **μικρές** (που χωράνε στην κύρια μνήμη) ή **μεγάλες ΒΔ**

10

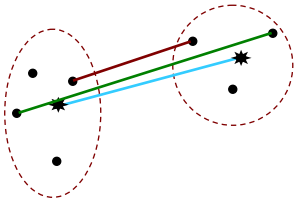
Προσεγγίσεις Συσταδοποίησης



11

Απόσταση μεταξύ δύο Συστάδων

- **Απόσταση απλού συνδέσμου** (single link): η ελάχιστη απόσταση μεταξύ στοιχείων των συστάδων
- **Απόσταση πλήρους Συνδέσμου** (complete link): η μέγιστη απόσταση μεταξύ στοιχείων των συστάδων
- **Μέση απόσταση**: η μέση απόσταση μεταξύ στοιχείων των συστάδων
- **Centroid απόσταση**: απόσταση μεταξύ των κέντρων βάρους (centroids) των συστάδων



Χαρακτηριστικές τιμές μιας συστάδας N στοιχείων

$$centroid = C_m = \frac{\sum_{i=1}^{N}(t_{mi})}{N}$$

$$radius = R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{mi} - C_m)^2}{N}}$$

$$diameter = D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(t_{mi} - t_{mj})^2}{(N)(N-1)}}$$

12

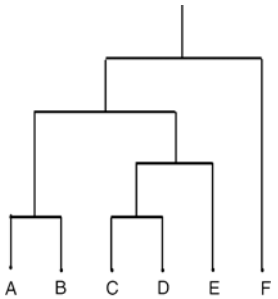
Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - **Ιεραρχικοί Αλγόριθμοι**
  - Διαμεριστικοί Αλγόριθμοι
  - Γενετικοί Αλγόριθμοι
  - Συσταδοποίηση Μεγάλων ΒΔ

13

Ιεραρχική Συσταδοποίηση

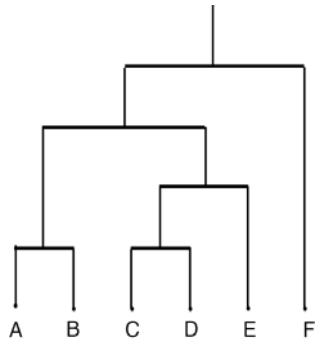
- Οι συστάδες δημιουργούνται σε επίπεδα
  - Κάθε επίπεδο αντιπροσωπεύει ένα σύνολο από συστάδες
- **Συσσωρευτικοί αλγόριθμοι** (agglomerative)
  - Αρχικά κάθε στοιχείο είναι μία συστάδα
  - Επαναληπτικά οι συστάδες συγχωνεύονται
  - Προσέγγιση bottom-up
- **Διαιρετικοί αλγόριθμοι** (divisive)
  - Αρχικά όλα τα στοιχεία σε μία συστάδα.
  - Οι μεγάλες συστάδες προοδευτικά διαιρούνται.
  - Προσέγγιση top-down



14

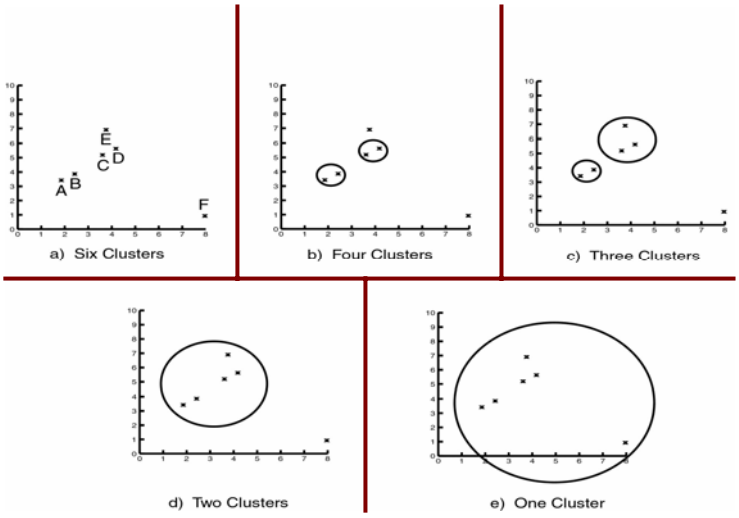
Δενδρόγραμμα

- **Δενδρόγραμμα** (dendrogram): μία δενδρική δομή δεδομένων η οποία επιδεικνύει τις ιεραρχικές τεχνικές συσταδοποίησης.
- Κάθε επίπεδο δείχνει τις συστάδες εκείνου του επιπέδου.
  - Φύλλα – κάθε στοιχείο αποτελεί ξεχωριστή συστάδα
  - Ρίζα – όλα τα στοιχεία αποτελούν μία συστάδα
- Μία συστάδα στο επίπεδο i είναι η ένωση των συστάδων-παιδιών στο επίπεδο i+1.



15

Επίπεδα της Συσταδοποίησης



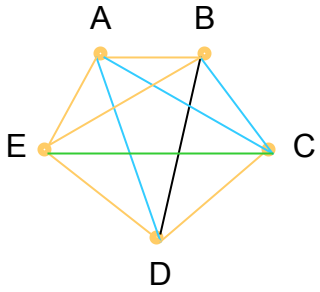
16

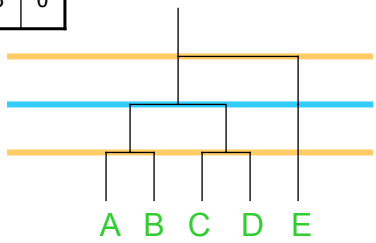
Παράδειγμα Συσσώρευσης

Μήτρα γειτνίασης

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Γράφος αποστάσεων





Κατώφλι απόστασης

1 2 3 4 5

17

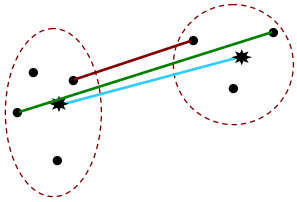
Συσσωρευτικός Αλγόριθμος

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
Output:
DE // Dendrogram represented as a set of ordered triples.
Agglomerative Algorithm:
$d = 0$;
$k = n$;
$K = \{\{t_1\}, ..., \{t_n\}\}$;
$DE = \{< d, k, K >\}$; // Initially dendrogram contains each element in its own cluster.
repeat
$oldk = k$;
$d = d + 1$;
A_d = Vertex adjacency matrix for graph with threshold distance of d;
$< k, K >= NewClusters(A_d, D)$;
if $oldk \neq k$ **then**
$DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.
until $k = 1$

18

Προσεγγίσεις Συσσωρευτικού Αλγόριθμου

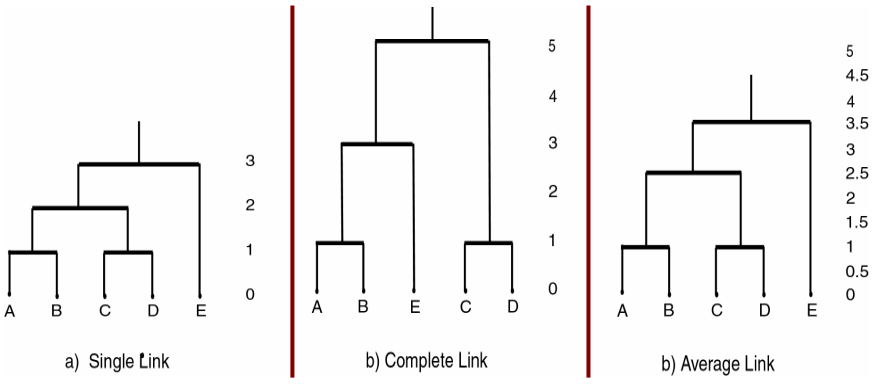
Με βάση την τεχνική που χρησιμοποιείται για τον καθορισμό της απόστασης μεταξύ δύο συστάδων



- **Τεχνική Απλού Συνδέσμου** (single link)
  - αναζητά συνεκτικές συνιστώσες στο γράφο αποστάσεων
  - ονομάζεται και **τεχνική συσταδοποίησης πλησιέστερου γείτονα** (nearest neighbor)
  - Παραλλαγή: με χρήση **δένδρου ελάχιστης ζεύξης** (Minimum Spanning Tree – MST)
- **Τεχνική Πλήρους Συνδέσμου** (complete link)
  - αναζητά κλίκες στο γράφο αποστάσεων
  - Παραλλαγή: **τεχνική συσταδοποίησης απώτατου γείτονα** (farthest neighbor)
- **Τεχνική Μέσου Συνδέσμου** (average link)

19

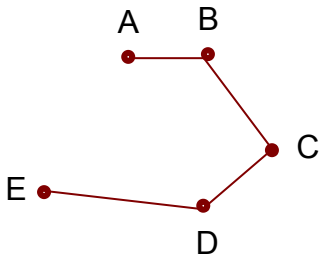
Εναλλακτικές Τεχνικές Συσσώρευσης

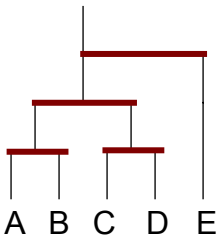


20

Συσσώρευση βασισμένη σε MST

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0





21

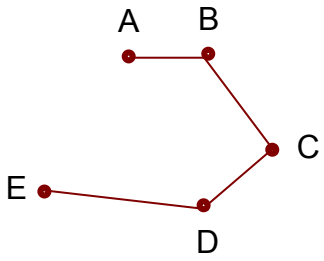
Συσσωρευτικός Αλγόριθμος Απλού Συνδέσμου βασισμένος σε MST

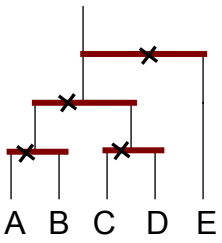
Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
Output:
DE // Dendrogram represented as a set of ordered triples.
MST Single Link Algorithm:
$d = 0$;
$k = n$;
$K = \{\{t_1\}, ..., \{t_n\}\}$;
$DE =< d, k, K >$; // Initially dendrogram contains each element in its own cluster.
$M = MST(A)$;
repeat
$oldk = k$;
$K_i, K_j =$ **two clusters closest together in MST;**
$K = K - \{K_i\} - \{K_j\} \cup \{K_i \cup K_j\}$;
$k = oldk - 1$;
$d = dis(K_i, K_j)$;
$DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.
$dis(K_i, K_j) = \infty$;
until $k = 1$

22

Διαίρεση βασισμένη σε MST

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0





23

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - **Διαμεριστικοί Αλγόριθμοι**
  - Γενετικοί Αλγόριθμοι
  - Συσταδοποίηση Μεγάλων ΒΔ

24

Συσταδοποίηση με Διαμέριση

- Μη ιεραρχική
- Δημιουργεί τις συστάδες σε ένα βήμα μόνο.
- Εφόσον υπάρχει μόνο ένα σύνολο συστάδων στην έξοδο, ο χρήστης πρέπει να εισάγει τον επιθυμητό αριθμό των συστάδων, k.
- Συνήθως χειρίζεται στατικά σύνολα.

- Πρόβλημα: οι πιθανοί συνδυασμοί n στοιχείων σε k συστάδες είναι ένας πολύ μεγάλος αριθμός (π.χ. $>10^{10}$ για n=19, k=4)
  - Αναγκαστικά, η αναζήτηση γίνεται σε ένα μικρό υποσύνολο των πιθανών λύσεων

25

Διαμεριστικοί Αλγόριθμοι

- Τεχνική βασισμένη σε **Δένδρο Ελάχιστης Ζεύξης** (MST)
- **Τετραγωνικού Σφάλματος** (squared error)
- **K-Μέσων** (K-means)
- **Πλησιέστερου Γείτονα** (nearest neighbor)
- **PAM** (partitioning around medoids – διαμερισμός γύρω από medoids)
- Τεχνική βασισμένη σε **Γενετικούς Αλγορίθμους**
- Τεχνική βασισμένη σε **Νευρωνικά Δίκτυα**
- ...

26

Αλγόριθμος MST

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
k // Number of desired clusters.
Output:
f // Mapping represented as a set of ordered pairs.
Partitional MST Algorithm:
$M = MST(A)$;
identify inconsistent edges in M;
remove $k-1$ inconsistent edges;
create output representation;

Πολυπλοκότητα $O(n^2)$

$O(n^2)$ για τη δημιουργία του MST + $O(k^2)$ για τα 3 τελευταία βήματα

27

Αλγόριθμος Squared Error

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
k // Number of desired clusters.
Output:
K // Set of clusters.
Squared Error Algorithm:
assign each item t_i to a cluster;
calculate center for each cluster;
repeat
assign each item t_i to the cluster which has the closest center ;
calculate new center for each cluster;
calculate squared error;
until the difference between successive squared errors is below a threshold;

Στόχος: η ελαχιστοποίηση του Τετραγωνικού Σφάλματος

$$se_{K_i} = \sum_{j=1}^{m} \|t_{ij} - C_k\|^2$$

$$se_K = \sum_{j=1}^{k} se_{K_j}$$

Πολυπλοκότητα O(tkn)

όπου t το πλήθος των επαναλήψεων

28

Συσταδοποίηση K-Means

- Το αρχικό σύνολο συστάδων επιλέγεται τυχαία.
- Επαναληπτικά, τα στοιχεία μετακινούνται μεταξύ συνόλων συστάδων μέχρι να φτάσουμε το επιθυμητό σύνολο.
- Επιτυγχάνεται υψηλός βαθμός ομοιότητας μεταξύ των στοιχείων μίας συστάδας.
- Δεδομένης μίας συστάδας $K_i=\{t_{i1},t_{i2},\ldots,t_{im}\}$, ο **μέσος της συστάδας** είναι $m_i = (1/m)(t_{i1} + \ldots + t_{im})$
  - Ο μέσος της συστάδας ταυτίζεται με το κέντρο βάρους

29

Παράδειγμα K-Means (σε 1 διάσταση)

- Δίνεται: {2, 4, 10, 12, 3, 20, 30, 11, 25}, k=2
- Τυχαία επιλέγουμε, έστω $m_1=3$, $m_2=4$
- 1η επανάληψη: $K_1=\{2, 3\}$, $K_2=\{4, 10, 12, 20, 30, 11, 25\}$, $m_1=2.5$, $m_2=16$
- 2η επανάληψη: $K_1=\{2, 3, 4\}$, $K_2=\{10, 12, 20, 30, 11, 25\}$, $m_1=3$, $m_2=18$
- 3η επανάληψη: $K_1=\{2, 3, 4, 10\}$, $K_2=\{12, 20, 30, 11, 25\}$, $m_1=4.75$, $m_2=19.6$
- 4η επανάληψη: $K_1=\{2, 3, 4, 10, 11, 12\}$, $K_2=\{20, 30, 25\}$, $m_1=7$, $m_2=25$
- 5η επανάληψη: δεν αλλάζει τίποτα. Τέλος

30

Αλγόριθμος K-Means

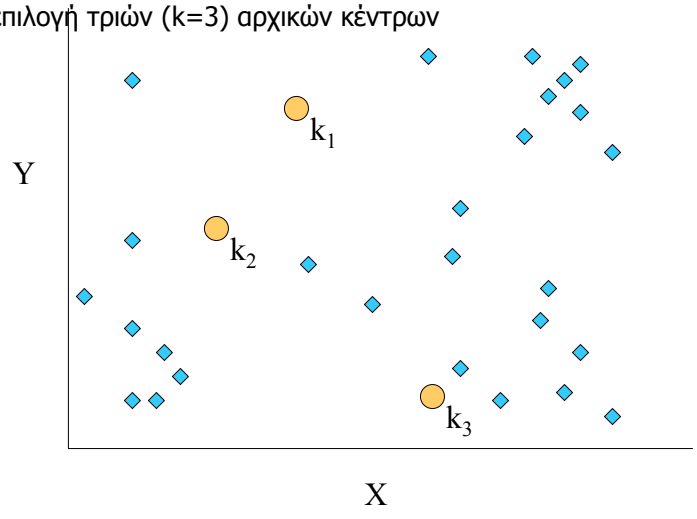
Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
k // Number of desired clusters.
Output:
K // Set of clusters.
K-Means Algorithm:
assign initial values for means $m_1, m_2, ..., m_k$;
repeat
assign each item t_i to the cluster which has the closest mean ;
calculate new mean for each cluster;
until convergence criteria is met;

Πολυπλοκότητα O(tkn)
όπου t το πλήθος των επαναλήψεων

31

Παράδειγμα K-Means (σε 2 διαστάσεις)

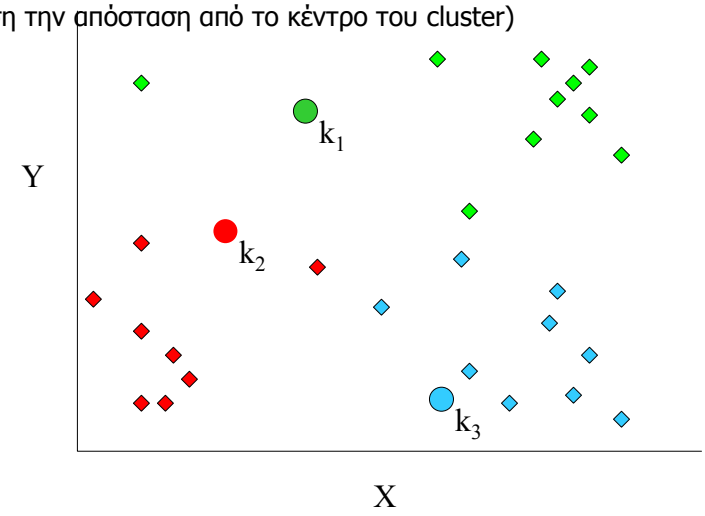
- Τυχαία επιλογή τριών (k=3) αρχικών κέντρων



32

Παράδειγμα K-means, 1η επανάληψη

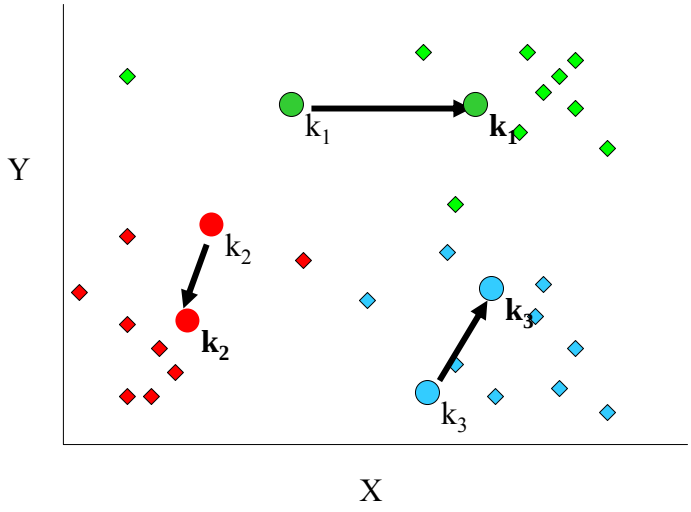
- Εκχώρηση κάθε στοιχείου στο πλησιέστερό του cluster (με βάση την απόσταση από το κέντρο του cluster)



33

Παράδειγμα K-means, 1η επανάληψη

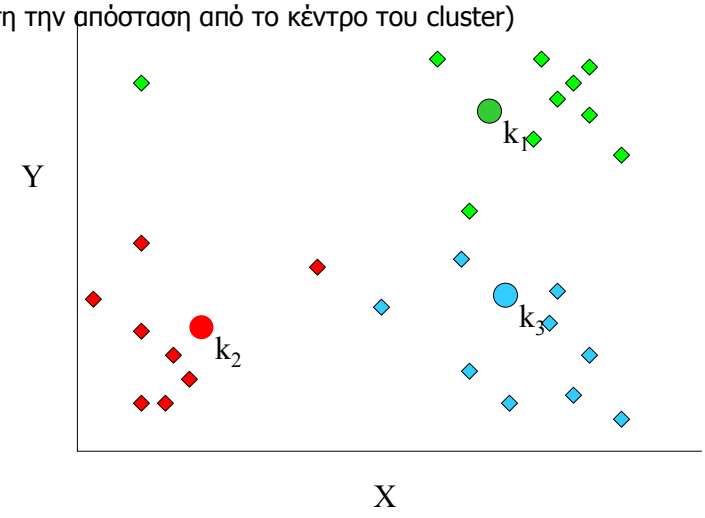
- Επανυπολογισμός του νέου κέντρου βάρους του κάθε cluster



34

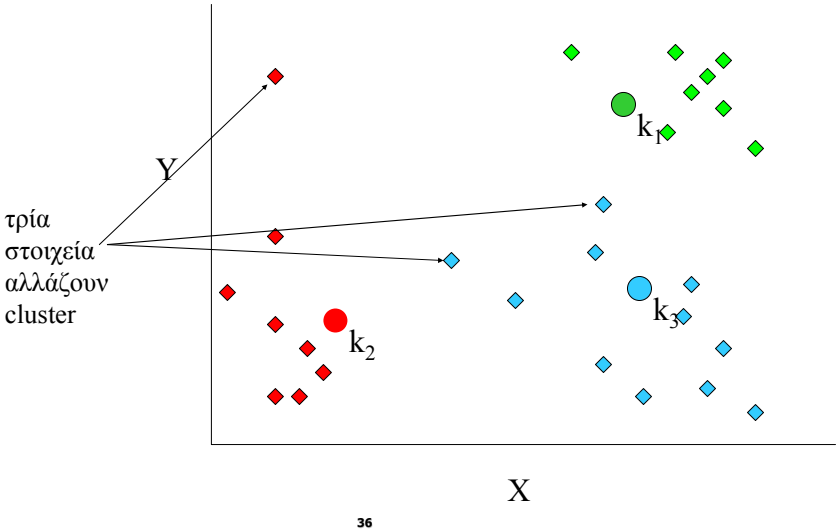
Παράδειγμα K-means, 2η επανάληψη

- Εκχώρηση κάθε στοιχείου στο πλησιέστερό του cluster (με βάση την απόσταση από το κέντρο του cluster)



35

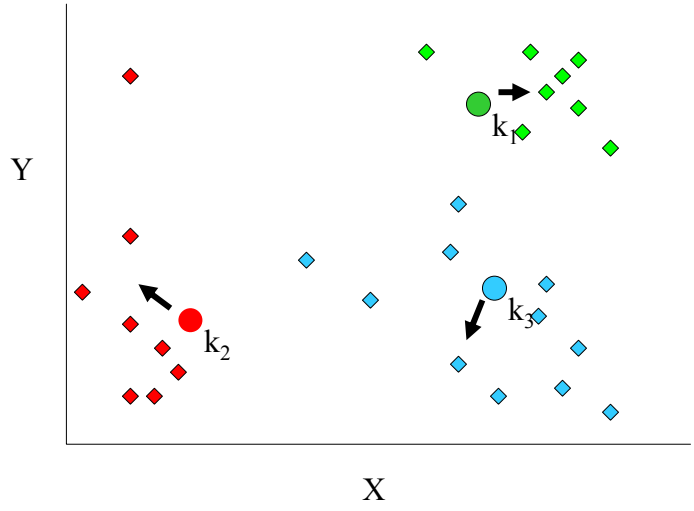
Παράδειγμα K-means, 2η επανάληψη



36

Παράδειγμα K-means, 2η επανάληψη

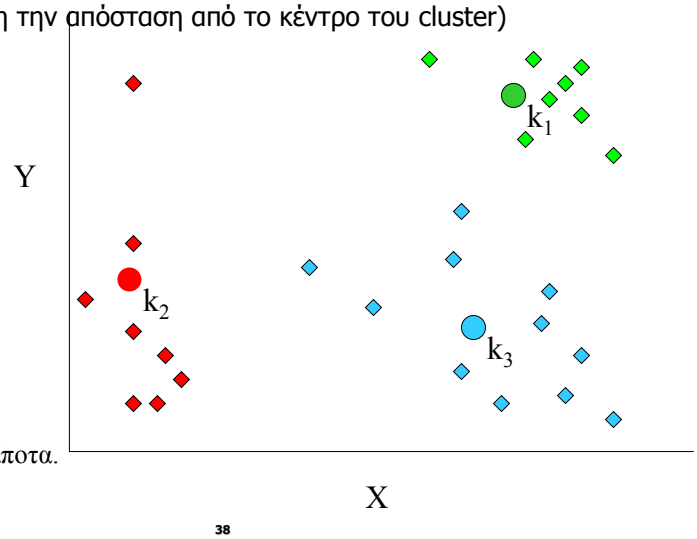
- Επανυπολογισμός του νέου κέντρου βάρους του κάθε cluster



37

Παράδειγμα K-means, 2η επανάληψη

- Εκχώρηση κάθε στοιχείου στο πλησιέστερό του cluster (με βάση την απόσταση από το κέντρο του cluster)



Δεν αλλάζει τίποτα.
Άρα, τέλος !

38

Πλησιέστερος Γείτονας

- Τα στοιχεία συγχωνεύονται επαναληπτικά μέσα σε συστάδες που βρίσκονται πιο κοντά.
- Αυξητική προσέγγιση
- Ένα κατώφλι, t, χρησιμοποιείται για να καθορίσει εάν ένα στοιχείο θα ενταχθεί σε μία από τις υπάρχουσες συστάδες ή εάν θα δημιουργηθεί μία νέα συστάδα.

39

Αλγόριθμος Πλησιέστερου Γείτονα

Input:
  $D = \{t_1, t_2, ..., t_n\}$ // Set of elements
  A // Adjacency matrix showing distance between elements.
Output:
  K // Set of clusters.
Nearest Neighbor Algorithm:
  $K_1 = \{t_1\}$;
  $K = \{K_1\}$;
  $k = 1$;
  for $i = 1$ *to* n do
    find the t_m in some cluster K_m in K such that $dis(t_i, t_m)$ is the smallest;
    if $dis(t_i, t_m) \leq t$ then
      $K_m = K_m \cup t_i$
    else
      $k = k + 1$;
      $K_k = \{t_i\}$;

Πολυπλοκότητα $O(n^2)$

40

PAM (k-medoids)

- **Διαμέριση γύρω από Medoids (PAM)**
- Χειρίζεται καλά τα ακραία σημεία (σε αντίθεση με τον K-Means).
- Η διάταξη της εισόδου δεν επηρεάζει τα αποτελέσματα (σε αντίθεση με τον K-Means).
- Κάθε συστάδα αναπαρίσταται από ένα αντιπροσωπευτικό στοιχείο, το οποίο καλείται **medoid**(*).
  - Προσοχή: το medoid ΔΕΝ είναι το κέντρο βάρους της συστάδας (ή ο μέσος, στην ορολογία του K-means) αλλά ένα από τα στοιχεία της
- Το αρχικό σύνολο των k medoids επιλέγεται τυχαία.

- (*) 'τεχνικός' όρος που προέρχεται από τον όρο multivariate median (πολυμεταβλητός μέσος)

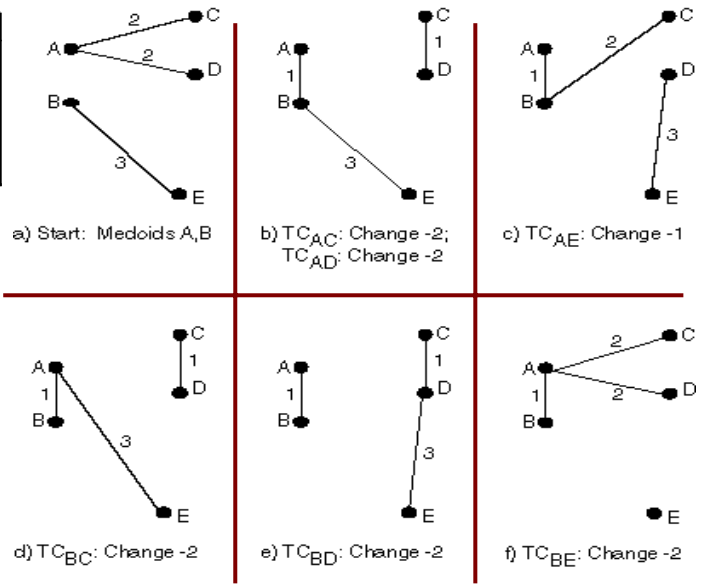
41

Παράδειγμα PAM

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

TC_{ih}: το κόστος αντικατάστασης του μέσου i από τον μη-μέσο j

Στόχος είναι να βρεθεί η αντικατάσταση με το ελάχιστο κόστος



42

Υπολογισμός Κόστους TC_{ih}

- Σε κάθε βήμα του αλγορίθμου, οι μέσοι μεταβάλλονται εάν το **συνολικό κόστος βελτιώνεται**.
- C_{jih} – αλλαγή κόστους για ένα στοιχείο t_j που σχετίζεται με την αντικατάσταση του μέσου t_i από τον μη-μέσο t_h.
- 4 περιπτώσεις προς εξέταση:

1. $t_j \in K_i$, but $\exists$ another medoid t_m *where* $dis(t_j, t_m) \leq dis(t_j, t_h)$
2. $t_j \in K_i$, but $dis(t_j, t_h) \leq dis(t_j, t_m) \forall$ *other medoids* t_m;
3. $t_j \in K_m$, $\notin K_i$, and $dis(t_j, t_m) \leq dis(t_j, t_h)$; and
4. $t_j \in K_m$, $\notin K_i$, but $dis(t_j, t_h) \leq dis(t_j, t_m)$.

43

Αλγόριθμος PAM

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
k // Number of desired clusters.
Output:
K // Set of clusters.
PAM Algorithm:

```
arbitrarily select k medoids from D;
repeat
    for each t_h not a medoid do
        for each medoid t_i do
            calculate TC_ih;
    find i, h where TC_ih is the smallest;
    if TC_ih < 0 then
        replace medoid t_i with t_h;
until TC_ih ≥ 0;
for each t_i ∈ D do
    assign t_i to K_j where dis(t_i, t_j) is the smallest over all medoids;
```

Πολυπλοκότητα $O(tn(n-k)^2)$
όπου t το πλήθος των επαναλήψεων

44

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - Διαμεριστικοί Αλγόριθμοι
  - **Γενετικοί Αλγόριθμοι**
  - Συσταδοποίηση Μεγάλων ΒΔ

45

Παράδειγμα Γενετικού Αλγορίθμου

- {A, B, C, D, E, F, G, H}
- Τυχαία επιλέγουμε αρχική λύση:

  {A, C, E} {B, F} {D, G, H} ή

  10101000, 01000100, 00010011
- Υποθέτουμε διασταύρωση στο σημείο 4 και επιλέγουμε τα 1 και 3 ως γονείς:

  10100011, 01000100, 00011000
- Ποια πρέπει να είναι τα κριτήρια τερματισμού;

46

Αλγόριθμος GA

Input:
$D = \{t_1, t_2, ..., t_n\}$ // *Set of elements*
k // *Number of desired clusters.*
Output:
K // *Set of clusters.*
GA Clustering Algorithm:
randomly create an initial solution;
repeat
use crossover to create a new solution;
until *termination criteria is met;*

47

Περιεχόμενα

- Γενική εικόνα προβλήματος συσταδοποίησης
- Τεχνικές Συσταδοποίησης
  - Ιεραρχικοί Αλγόριθμοι
  - Διαμεριστικοί Αλγόριθμοι
  - Γενετικοί Αλγόριθμοι
  - **Συσταδοποίηση Μεγάλων ΒΔ**

48

Συσταδοποίηση Μεγάλων ΒΔ

- Οι περισσότεροι αλγόριθμοι συσταδοποίησης προϋποθέτουν μία μεγάλη δομή δεδομένων που βρίσκεται στην κύρια μνήμη.
- Η συσταδοποίηση θα μπορούσε να εφαρμοστεί πρώτα σε ένα δείγμα της ΒΔ και μετά σε ολόκληρη τη ΒΔ.
- Αλγόριθμοι
  - BIRCH
  - DBSCAN
  - CURE

49

Επιθυμητά Χαρακτηριστικά για Μεγάλες ΒΔ

- Ένα πέρασμα (το πολύ) της ΒΔ
- Απευθείας επεξεργασία
- Με δυνατότητα διακοπής, τέλους, επανεκίνησης
- Αυξητικό
- Να δουλεύει με περιορισμένη κύρια μνήμη
- Διαφορετικές Τεχνικές Περασμάτων (π.χ. δειγματοληψία)
- Επεξεργασία κάθε εγγραφής μία μόνο φορά

50

BIRCH

- Σταθμισμένη Επαναληπτική Μείωση και Συσταδοποίηση με την χρήση Ιεραρχιών
- Αυξητική, ιεραρχική, ένα πέρασμα
- Σώζουμε πληροφορίες σχετικές με την συσταδοποίηση σε ένα δέντρο
- Κάθε καταχώρηση του δέντρου περιέχει πληροφορίες για μία συστάδα
- Καινούργιοι κόμβοι εισάγονται στην πιο κοντινή καταχώρηση του δέντρου

51

Ιδιότητα Συσταδοποίησης

- CT Τριάδα: $(N, \vec{LS}, SS)$
  - N: Αριθμός σημείων στη συστάδα
  - $\vec{LS}$: Άθροισμα σημείων στην συστάδα
  - SS: Άθροισμα τετραγώνων σημείων της συστάδας
- CF Δέντρο
  - Σταθμισμένο Δέντρο Αναζήτησης
  - Ο κόμβος έχει μία CF τριάδα για κάθε παιδί
  - Το φύλλο αναπαριστά τη συστάδα και έχει τιμή CF για κάθε υποσυστάδα μέσα σε αυτή.
  - Η υποσυστάδα έχει μέγιστη διάμετρο

52

Αλγόριθμος BIRCH

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
T // Threshold for CF tree construction.
Output:
K // Set of clusters.
BIRCH Clustering Algorithm:

```
for each t_i ∈ D do
    determine correct leaf node for t_i insertion;
    if threshold condition is not violated then
        add t_i to cluster and update CF triples;
    else
        if room to insert t_i then
            insert t_i as single cluster and update CF triples;
        else
            split leaf node and redistribute CF features;
```

53

Βελτίωση Συστάδων

1. Create initial CF tree using Algorithm. If there is insufficient memory to construct the CF tree with a given threshold, the threshold value is increased, and a new smaller CF tree is constructed.
2. Apply another global clustering approach applied to the leaf nodes in the CF tree. Here each leaf node is treated as a single point for clustering.
3. The last phase (which is optional) reclusters all points by placing them in the cluster which has the closest centroid.

54

DBSCAN

- Χωρική Συσταδοποίηση Εφαρμογών με Θόρυβο με βάση την Πυκνότητα
- Τα ακραία σημεία δεν θα έχουν ως αποτέλεσμα την δημιουργία μίας συστάδας.
- Είσοδος
  - **MinPts** – ελάχιστος αριθμός σημείων στη συστάδα
  - **Eps** – για ˙κάθε σημείο σε μία συστάδα θα πρέπει να υπάρχει ένα άλλο σημείο σε αυτό με μικρότερη από αυτή την απόσταση μακρυά.

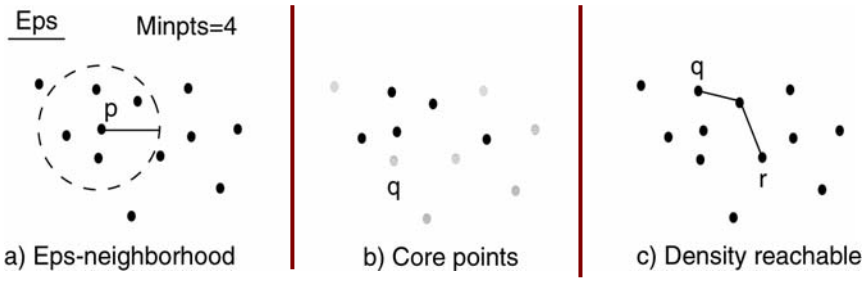
55

DBSCAN 'Εννοιες της Πυκνότητας

- **Eps-γειτονιά**: Σημεία μέσα σε Eps απόσταση από σημείο.
- **Σημείο Πυρήνα**: Eps-γειτονιά αρκετά πυκνή (MinPts)
- **Απευθείας density-reachable**: 'Ενα σημείο p είναι απευθείας density-reachable από ένα σημείο q εάν η απόσταση είναι μικρή (Eps) και το q είναι ένα σημείο του πυρήνα.
- **Density-reachable**: 'Ενα σημείο είναι density-reachable από ένα άλλο σημείο εάν υπάρχει ένα μονοπάτι από το ένα στο άλλο το οποίο αποτελείται μόνο από σημεία πυρήνες.

56

Έννοιες της Πυκνότητας



57

DBSCAN Αλγόριθμος

Input:
$D = \{t_1, t_2, ..., t_n\}$ //Set of elements.
$MinPts$ // Number of points in cluster.
Eps // Maximum distance for density measure.
Output:
$K = \{K_1, K_2, ..., K_k\}$ //Set of clusters.
DBSCAN Algorithm:
$k = 0$; // Initially there are no clusters.
for $i = 1$ *to* n **do**
if t_i **is not in a cluster then**
$X = \{t_j \mid t_j$ **is density-reachable from** $t_i\}$;
if X **is a valid cluster then**
$k = k + 1$;
$K_k = X$;

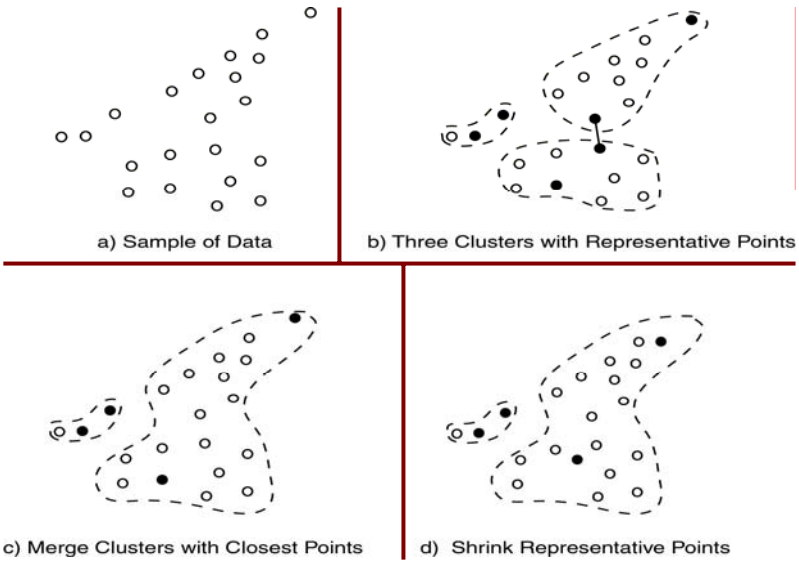
58

CURE

- Συσταδοποίηση με τη χρήση αντιπροσώπων.
- Χρήση πολλών σημείων για την αναπαράσταση μίας συστάδας αντί μόνο ένα.
- Τα σημεία θα είναι αρκετά διασκορπισμένα.

59

Προσέγγιση CURE



60

Αλγόριθμος CURE

Input:
$D = \{t_1, t_2, ..., t_n\}$ //Set of elements.
k // Desired number of clusters.
Output:
Q //Heap containing one entry for each cluster.
CURE Algorithm:

```
T = build(D);              // Put each point in Tree
Q = heapify(D);            // Initially build heap with one entry per item;
repeat
    u = min(Q);
    delete(Q, u.close);
    w = merge(u, v);
    delete(T, u);
    delete(T, v);
    insert(T, w);
    for each x ∈ Q do
        x.close = find closest cluster to x;
        if x is closest to w then
            w.close = x;
    insert(Q, w);
until number of nodes in Q is k;
```

61

CURE για μεγάλες ΒΔ

1. Obtain a sample of the database.
2. Partition the sample into p partitions.
3. Partially cluster the points in each partition.
4. Remove outliers based on size of cluster.
5. Completely cluster all data in the sample (representatives).
6. Cluster entire database on disk using c points to represent each cluster. An item in the database is placed in the cluster which has the closest representative point to it.

62

Σύνοψη

- Συσταδοποίηση: η εύρεση ομάδων μεταξύ των δεδομένων ενός συνόλου
  - με βάση ένα μέτρο απόστασης
- Τεχνικές:
  - Ιεραρχικές (συσσωρευτικές/διαιρετικές, απλού/πλήρους/μέσου συνδέσμου)
  - Διαμεριστικές (με πιο δημοφιλή τον αλγόριθμο Apriori)
  - Άλλες (βασισμένες στην πυκνότητα, σε γενετικούς αλγορίθμους, παράλληλες τεχνικές κ.α.)

63

Σύγκριση Τεχνικών Συσταδοποίησης

Algorithm	Type	Space	Time	Notes
Single Link	Hierarchical	$O(n^2)$	$O(kn^2)$	Not incremental
Average Link	Hierarchical	$O(n^2)$	$O(kn^2)$	Not incremental
Complete Link	Hierarchical	$O(n^2)$	$O(kn^2)$	Not incremental
MST	Hierarchical/ Partitional	$O(n^2)$	$O(n^2)$	Not incremental
Squared Error	Partitional	$O(n)$	$O(tkn)$	Iterative
K-Means	Partitional	$O(n)$	$O(tkn)$	Iterative, No categorical
Nearest Neighbor	Partitional	$O(n^2)$	$O(n^2)$	Incremental
PAM	Partitional	$O(tn^3)$ or $O(tkn^2)$	$O(n^2)$	Iterative
BIRCH	Partitional	$O(n)$	$O(n)$	CF-Tree; Incremental; Outliers
CURE	Mixed	$O(n^2lgn)$	$O(n)$	Heap; k-D tree; Incremental; Outliers
ROCK	Agglomerative	$O(n^2lgn)$	$O(n^2)$	Sampling; Categorical; Links
DBSCAN	Mixed	$O(n^2)$	$O(n^2)$	Sampling; Outliers

64