Curs 7:

Gruparea datelor (I)

Structura

- Gruparea datelor
  - Concepte de bază
  - Evaluarea calităţii grupării
- Algoritmi partiţionali
  - kMeans
  - Fuzzy cMeans
- Algoritmi ierarhici
  - Algoritmi aglomerativi
  - Algoritmi divizivi
- Algoritmi bazați pe estimarea densității datelor
- Algoritmi bazați pe modele probabiliste

Scopul grupării datelor (reminder)

Ce se cunoaşte?

- un set de date (nu neapărat structurate)
- O măsură de similaritate/disimilaritate între date (măsura e specifică problemei de rezolvat) pe baza căreia se construieşte o matrice de similaritate/disimilaritate

Ce se doreşte?

- Un model care descrie modul în care se grupează datele în clustere (grupuri) astfel încâte datele care aparţin aceluiaşi cluster sunt mai similare între ele decât cele care aparţin unor clustere diferite

Care este scopul final?

- Să se poată verifica dacă două date aparţin aceluiaşi cluster
- Să se determine clusterul de care aparţine o dată

Scopul grupării datelor (reminder)

Exemple:

- Customer segmentation = identificarea grupurilor de clienţi care au comportamente similare
- User profiles extraction = identificarea grupurilor de utilizatori ai unui serviciu web caracterizaţi prin comportamente similare
- Document clustering = identificarea unor grupuri de documente similare din punct de vedere al conţinutului
- Image segmentation = identificare unor regiuni omogene într-o imagine
- Gene expression analysis = gruparea genelor cu roluri similare

Gruparea permite:

- sumarizarea şi/sau vizualizarea datelor cu scopul de a înţelege mai bine datele

Particularităţi ale grupării datelor

Este un proces nesupervizat:

- Setul de antrenare conţine doar valori ale atributelor
- Eticheta clasei nu e cunoscută

Problema grupării este rău-definită:

- Identificarea grupurilor este dificilă
- Decizia poate avea caracter subiectiv

Câte clustere?

Şase clustere

Două clustere

patru clustere

[Kumar, 2004]

Concepte de bază

- Cluster = grup de date care sunt similare
- Matrice de (di)similaritate pt un set de n date = matrice nxn in care pe linia i coloana j se află valoarea similarităţii/disimilarităţii între data i şi data j
- Clustering (grupare) = proces de identificare a clusterelor
- Prototip (reprezentant)= "obiect" reprezentativ pentru datele dintr-un cluster
  - Centroid = media datelor dintr-un cluster – centroidul nu este neapărat un element din setul de date
  - Medoid = data din cluster care este cea mai apropiată de media clusterului – medoidul aparţine setului de date

- Raza clusterului = media distanţelor dintre datele din cluster şi prototipul acestuia
- Diametrul clusterului = distanţa (disimilaritatea) maximă dintre oricare două date ale clusterului

Tipuri de clustering

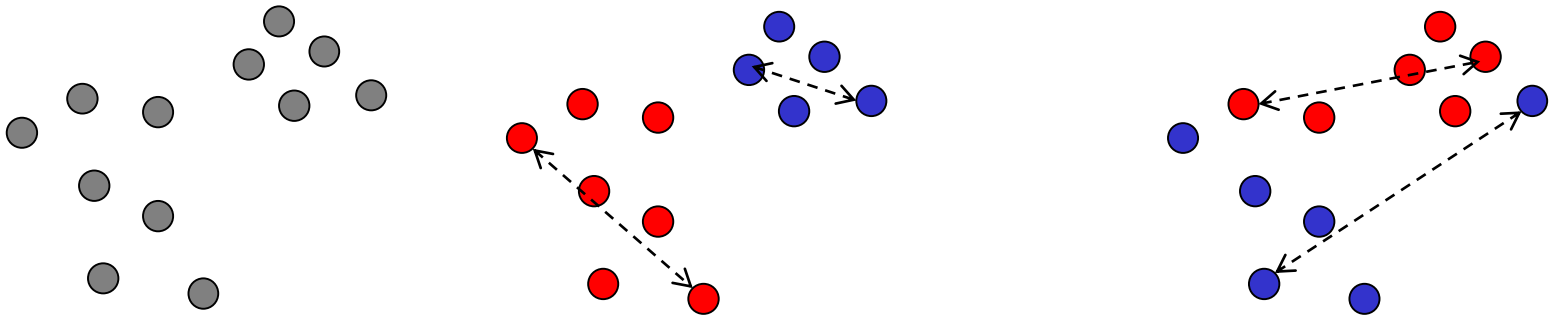
- Crisp vs fuzzy clustering
  - Crisp clustering = fiecare dată aparţine unui singur cluster
  - Fuzzy clustering = o dată poate aparţine mai multor clustere (grad de apartenenţă pentru fiecare cluster)

- Flat vs hierarchical clustering
  - Flat (partitional) clustering = rezultatul este un set de clustere (o partiţie)
  - Hierarchical clustering = rezultatul este o ierarhie de partiţii

- Variante de algoritmi
  - Algoritmi partiţionali (ex: kMeans, Fuzzy cMeans)
  - Algoritmi hierarhici (alg. aglomerativi, alg. divizivi)
  - Algoritmi bazaţi pe densitate (ex: DBSCAN)
  - Algoritmi bazați pe modele probabiliste (ex: EM = Expectation Maximization)

Măsuri de calitate

- Nu există un indicator unic pentru evaluarea calităţii unei grupări
- Cea mai comună abordare constă în estimarea:
  - Compacităţii clusterelor (variabilitate intra-cluster – ar trebui să fie mică)
  - Gradului de separare dintre datele apatţinând unor clustere diferite (variabilitate inter-cluster – ar trebui să fie mare)



Calitate bună

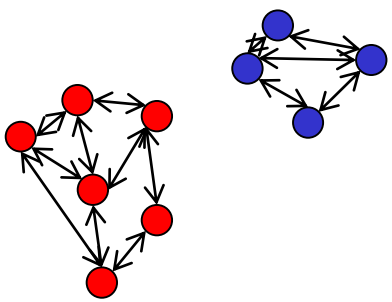
Calitate slabă

Măsuri de calitate

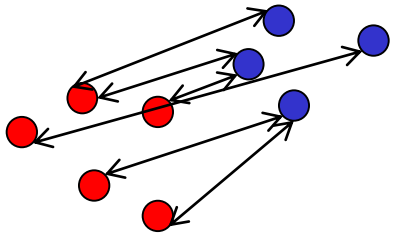
- Intra-cluster to inter-cluster distance ratio = Intra/Inter (valori mici corespund unei calităţi mai bune)
- Fie P setul de date ce aparţin aceluiaşi cluster şi Q=DxD-P (restul perechilor: o dată din pereche aparţine unui cluster iar cealaltă unui alt cluster)

$$Intra = \sum_{(x_i,x_j)\in P} d(x_i,x_j)/card(P)$$

$$Inter = \sum_{(x_i,x_j)\in Q} d(x_i,x_j)/card(Q)$$



Exemple de perechi de date utilizate în calculul distanţelor intra-cluster



Exemple de perechi de date utilizate în calculul distanţelor inter-cluster

Măsuri de calitate

- Silhouette coefficient

$$S_i = \frac{D\min_i^{out} - Davg_i^{in}}{\max\{D\min_i^{out}, Davg_i^{in}\}}$$

$$S = \frac{1}{n}\sum_{i=1}^{n} S_i$$

Obs:

- S ia valori în (-1,1)
- Valori mai mari indică o grupare mai bună

$Davg_i^{in} =$ media distantelor dintre x_i si celelalte date din clusterul lui x_i

$Davg_i^{j} =$ media distantelor dintre x_i si datele din clusterul j $(j \neq i)$

$D\min_i^{out} = \min_j D\,avg_i^{j}$

Algoritmi partiționali

- Input: set de date $D=\{x_1,x_2,\ldots,x_N\}$, K = nr de clustere
- Output: partiţie $P=\{C_1,C_2,\ldots,C_K\}$ a setului de date D

- Cel mai popular algoritmi partițional: kMeans (varianta Lloyd)

- Folosește ideea că prototipul fiecărui cluster este media aritmetică a elementelor din clusterul respectiv (denumit centroid)

- Este un algoritm euristic (bazat pe principiul greedy) având ca scop minimizarea distanțelor dintre date și centroizii corespunzători și care constă într-o prelucrare iterativă care
  - (re)asignează datele la clustere pe criteriul celui mai apropiat centroid – în general se utilizează distanța euclidiană
  - Recalculează centroizii folosind datele asignate la fiecare cluster

kMeans

- Input: set de date $D=\{x_1,x_2,\ldots,x_N\}$, K = nr de clustere
- Output: partiţie $P=\{C_1,C_2,\ldots,C_K\}$ a setului de date D

kMeans (D,k)

Initializare centroizi $c_1, c_2, \ldots, c_K$ (prin selecție aleatoare din setul de date)

repeat

- asignează fiecare dată din D la clusterul corespunzător celui mai apropiat centroid (în raport cu măsura de similaritate/disimilaritate folosită)
- actualizează fiecare centroid prin calculul mediei elementelor asignate clusterului

until <partiția este stabilă (la ultima iterație nu s-a modificat asignarea datelor la clustere) >

Obs:

- condiția de oprire poate fi bazată pe utilizarea unei măsuri de calitate (de exemplu varianța intra-cluster este suficient de mică)

kMeans

- Caracteristici
  - kMeans este o metodă bazată pe prototipuri care are ca scop minimizarea is sumei pătratelor erorilor (SSE) – distanţele dintre date şi centroizii corespunzători

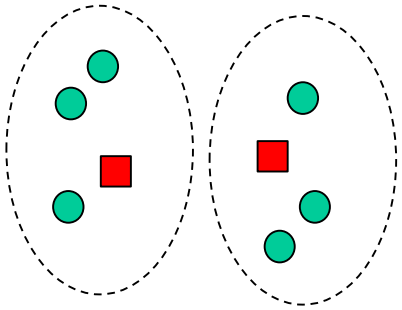
$$SSE(X, C) = \sum_{k=1}^{K} \sum_{x \in C_k} d^2(x, c_k) = \sum_{k=1}^{K} \sum_{x \in C_k} \sum_{j=1}^{n} (x_j - c_{kj})^2$$

(în cazul distanţei euclidiene)

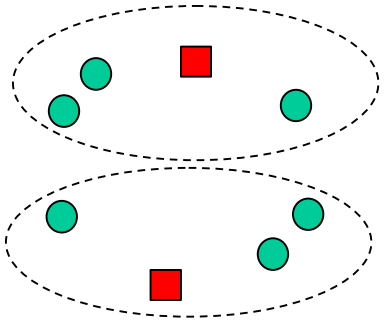
  - Complexitate: O(n*N*K*iteraţii) (n=nr de atribute, N=nr de date, K=nr de clustere)
  - Pre-procesare utilă: normalizare (eventual reducerea dimensiunii – ex: PCA)
  - Post-procesare utilă:
    - Eliminarea clusterelor mici (sunt considerate excepții)
    - Fragmentarea clusterelor caracterizate prin variabilitate mare
    - Reunirea clusterelor apropiate

kMeans

- Inițializarea centroizilor
  - Performanța algoritmului kMeans este influențată de inițializarea centroizilor
  - In varianta bazată pe selecție aleatoare o strategie de a determina o inițializare adecvată este de a aplica algoritmul repetat pornind de la valori diferite ale seed-ului corespunzător generatorului de valori aleatoare
  - kMeans++:
    - Primul centru se alege aleator din setul de date
    - Al doilea centru se alege folosind o distribuție de probabilitate care favorizează datele îndepărtate de primul centru
    - Al treilea centru se alege folosind o distribuție de probabilitate care favorizează datele îndepărtate de primele două centre
    - Se continuă procesul până la alegerea tuturor celor K centre



Inițializare favorabilă

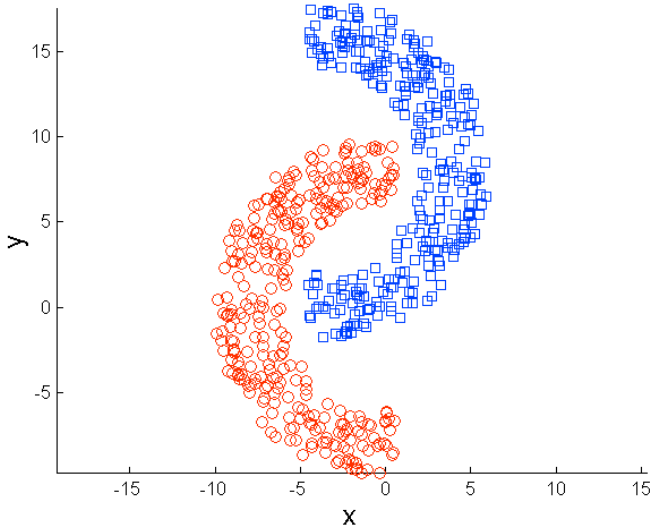


Inițializare nefavorabilă

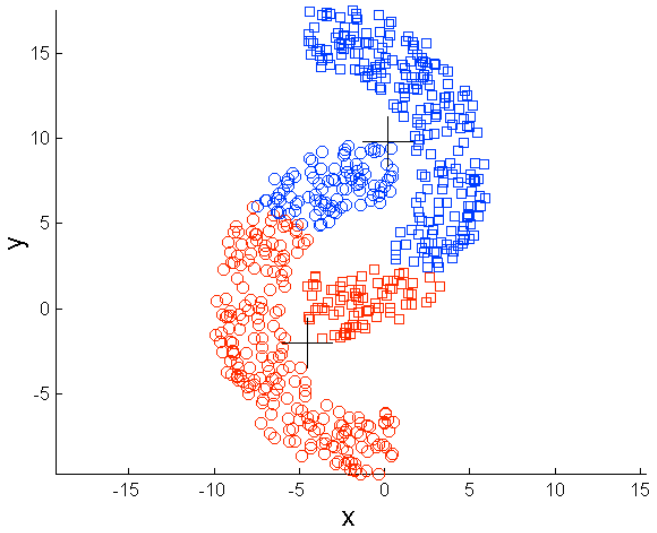
kMeans

Limite:

- Nu funcţionează bine dacă datele nu sunt “sferice” sau bine separate (distribuțiile de probabilitate corespunzătoare clusterelor nu sunt neapărat normale)
  - Soluţie: utilizarea altor abordări (e.g. clustering bazat pe densitate)



Clusterele reale



Rezultatul Kmeans

[from slides Kumar, 2004]

kMeans

Limite:

- Rezultatul poate fi ușor influențat de erorile din date sau de datele atipice
  - Soluţie: utilizarea medianei în locul mediei în construirea centroizii (reprezentanții clusterilor)

    - kMedian – se determină mediana la nivel de componentă (corespunde cazului în care se utilizează distanța Manhattan în criteriul de optimizat)
    - kMedoid – se determina elementul din setul de date care este cel mai apropiat de media clusterului (sau elementul din cluster care minimizează suma distanțelor dintre celelalte elemente ale clusterul și acel element); cel mai reprezentativ algoritm este Partitioning around Medoids – PAM (mai puțin eficient decât kMeans)

- Varianta bazată pe calculul mediei poate fi aplicată doar datelor numerice; pentru date nominale se poate utiliza varianta bazată pe mediană (kMedian) sau pe modă (se utilizează cea mai frecventă componentă – kMode)

kMeans

Limite: necesită cunoaşterea apriori a numărului de clustere

- Soluţii:
  - aplică algoritmul pt diferite valori ale lui K şi se selectează varianta care corespunde celor mai bune valori ale criteriilor de calitate
  - post-procesarea rezultatelor procesului de clustering prin partiţionarea clusterelor cu variabilitate mare şi reunirea clusterelor apropiate (ex: alg. ISODATA)

ISODATA

Idei principlale ale alg ISODATA

- Dacă dimensiunea unui cluster este mai mică decât Nmin (un parametru care specifică numărul minim de date dintr-un cluster) atunci clusterul ar trebui reunit cu cel mai apropiat alt cluster

- Dacă distanţa dintre două clustere (de exemplu distanţa dintre prototipurile clusterelor) este mai mică decât Dmin (un parametru care specifică distanța minimă dintre clustere) atunci clusterele ar trebui reunite
- Dacă varianţa unui cluster este mai mare decât Vmax şi numărul de date conţinute este mai mare decât 2*Nmin atunci clusterul poate fi divizat în două alte clustere:
  - Identifică atributul j pt care varianţa este maxmă
  - Din prototipul c_k sunt construite două alte prototipuri c' şi c'' prin înlocuirea valorii atributului j din c_k cu $c_k(j)$-b respectiv $c_k(j)$+b, r (b este un parametru setat de către utilizator)

Fuzzy cMeans

Ideea grupării fuzzy (soft):

- O dată nu aparţine unui singur cluster ci poate aparţine mai multor clustere (cu un anumit grad de apartenenţă pentru fiecare cluster)

- Rezultatul unei grupări fuzzy este o matrice de apartenență M de dimensiune NxK

(N= nr date, K= nr clustere);

M(i,j) = o valoare din [0,1] care corespunde gradului de apartenenţă a datei i la clusterul j

Obs: Fuzzy cMeans poate fi utilizată în segmentarea imaginilor

Fuzzy cMeans

Algoritm

- Initializare matrice de apartenenţă (M)
- **Repeat**
  - Calcul centroizi (c_k, k=1,…K)
  - Actualizare valori matrice de apartenenţă (m_{ij}, i=1,…,N, j=1,…,K

  until <matricea de apartenenţă nu se mai modifică>

Calculul centroizilor

$$c_j = \frac{\sum_{i=1}^{n} M_{ij}^{p} x_i}{\sum_{i=1}^{n} M_{ij}^{p}}, \qquad j = \overline{1, K}$$

p>1 is a parameter (e.g. p=2)

Obs: pentru a obține partiția, la finalul prelucrării fiecare dată este asignată la clusterul ce corespunde celui mai mare grad de apartenenţă

Calculul gradului de apartenenţă

$$M_{ij} = \frac{1}{\left\|x_i - c_j\right\|^{2/(p-1)} \sum_{k=1}^{K} 1/\|x_i - c_k\|^{2/(p-1)}}$$

$$i = \overline{1, n}, j = \overline{1, K}$$

Algoritmi ierarhici

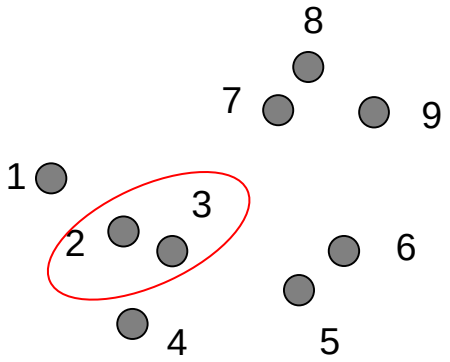
Obs: una dintre principalele limite ale algoritmilor partiționali e faptul că necesită cunoaşterea numărului de clustere.

Altă abordare: se construieşte o ierarhie de partiţii – conduce la o structură arborescentă numită dendrogramă

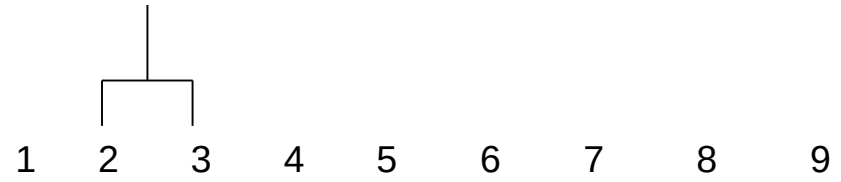
- In varianta bottom-up (metoda aglomerativă)
  - Se porneşte cu o partiţie de clustere ce conţin fiecare câte o singură dată (reprezintă frunze în arbore)
  - Se reunesc clusterele care sunt similare între ele – procesul de reunire se repetă până se ajunge la un singur cluster (reprezintă rădăcina arborelui)
- In varianta top-down (metoda divizivă)
  - Se porneşte cu o partiţie ce conţine un singur cluster (cu toate datele)
  - Se partiţionează clusterii mari aplicând o tehnica partiţională (ex: kMeans) – procesul continuă până când se ajunge la clustere ce conţin câte o singură dată.

Metoda aglomerativă

Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc

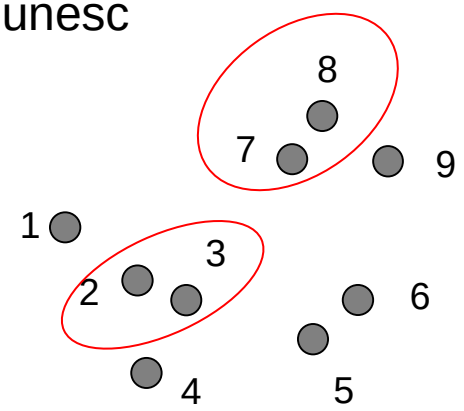


	1	2	3	4	5	6	7	8	9
1	0	2	3	4	7	8	6	8	10
2	2	0	1	2	4	6	7	8	9
3	3	1	0	2	3	5	6	8	9
4	4	2	2	0	3	6	9	10	11
5	7	4	3	3	0	1	4	6	5
6	8	6	5	6	1	0	3	4	3
7	6	7	6	9	4	3	0	1	2
8	8	8	8	10	6	4	1	0	2
9	10	9	9	11	5	3	3	2	0

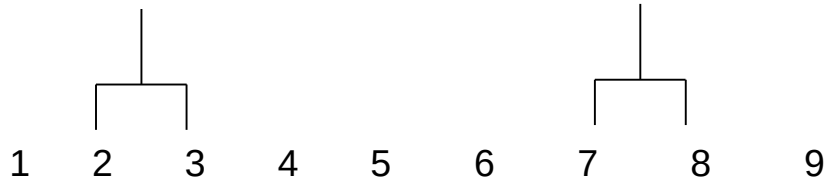


Metoda aglomerativă

Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc

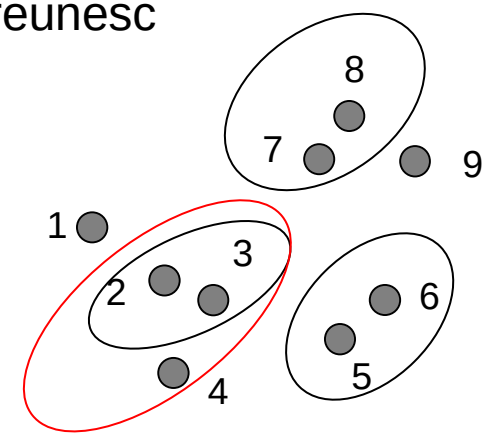


	1	2	3	4	5	6	7	8	9
1	0	2	3	4	7	8	6	8	10
2	2	0	1	2	4	6	7	8	9
3	3	1	0	2	3	5	6	8	9
4	4	2	2	0	3	6	9	10	11
5	7	4	3	3	0	1	4	6	5
6	8	6	5	6	1	0	3	4	3
7	6	7	6	9	4	3	0	1	2
8	8	8	8	10	6	4	1	0	2
9	10	9	9	11	5	3	3	2	0

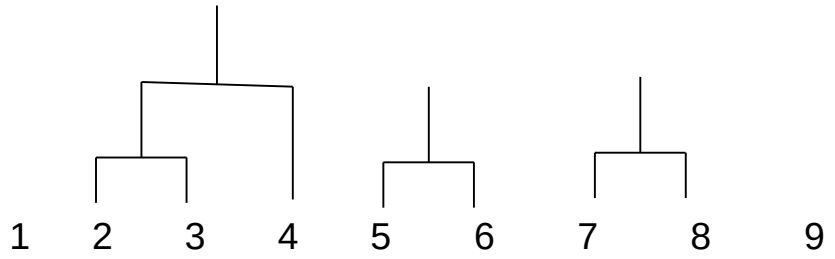


Metoda aglomerativă

Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc

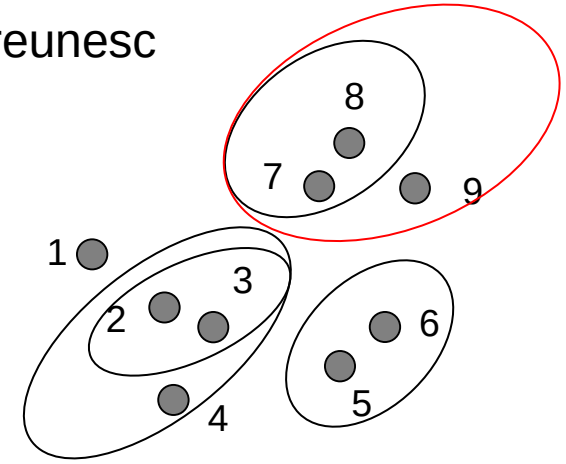


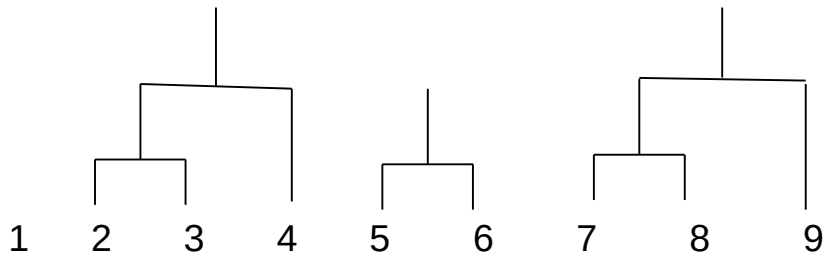
	1	2	3	4	5	6	7	8	9
1	0	2	3	4	7	8	6	8	10
2	2	0	1	2	4	6	7	8	9
3	3	1	0	2	3	5	6	8	9
4	4	2	2	0	3	6	9	10	11
5	7	4	3	3	0	1	4	6	5
6	8	6	5	6	1	0	3	4	3
7	6	7	6	9	4	3	0	1	2
8	8	8	8	10	6	4	1	0	2
9	10	9	9	11	5	3	3	2	0



Metoda aglomerativă

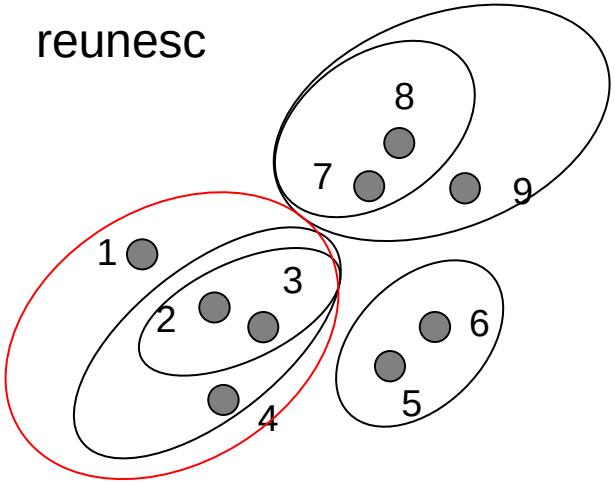
Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc

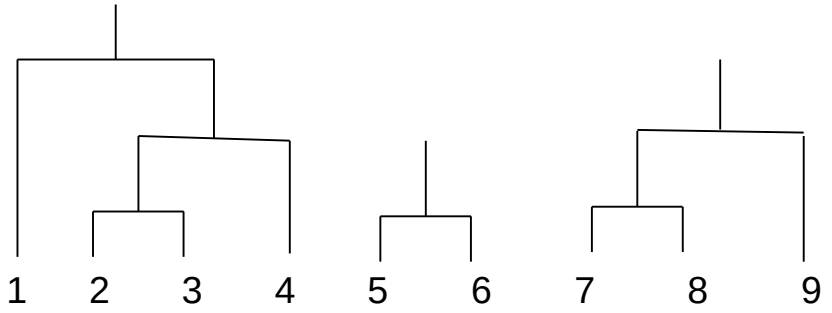




Metoda aglomerativă

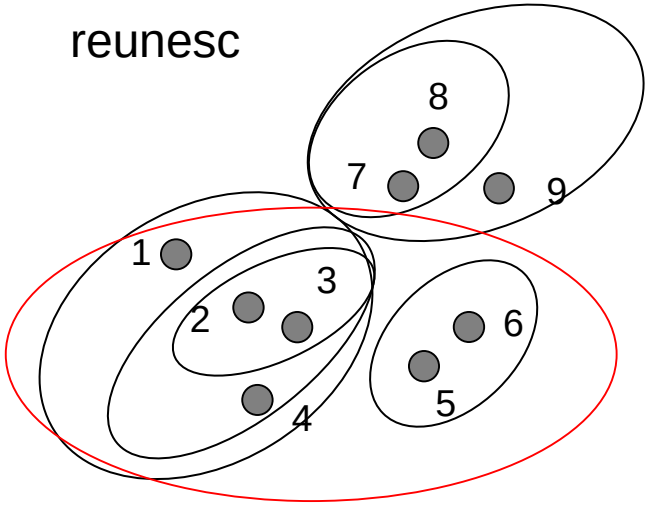
Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc

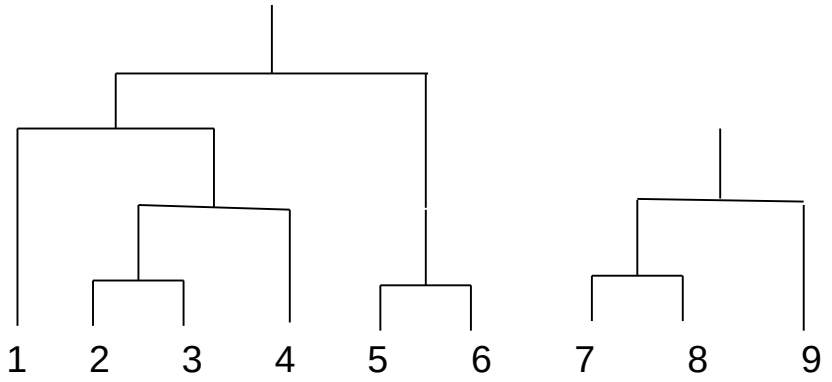




Metoda aglomerativă

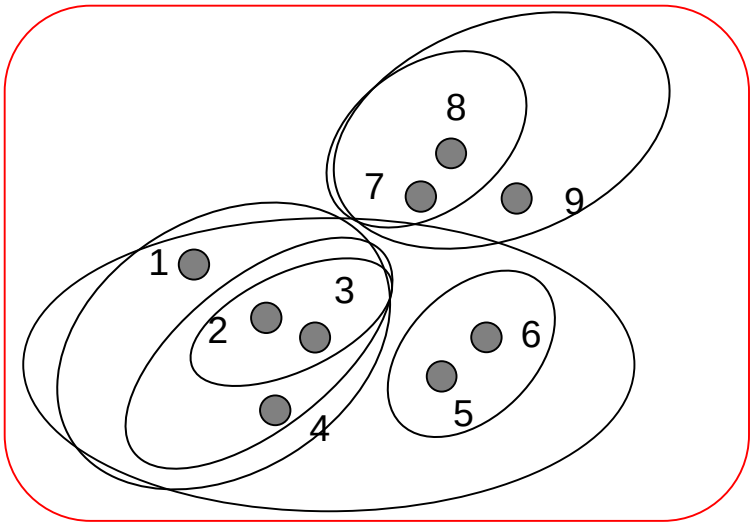
Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc



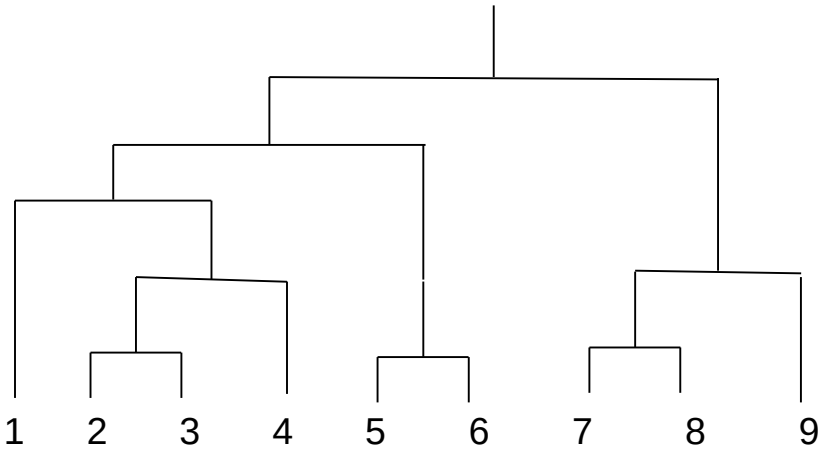


Metoda aglomerativă

Idee: se identifică la fiecare etapă care sunt cele mai similare clustere şi se reunesc



- Dendrograma rezultată

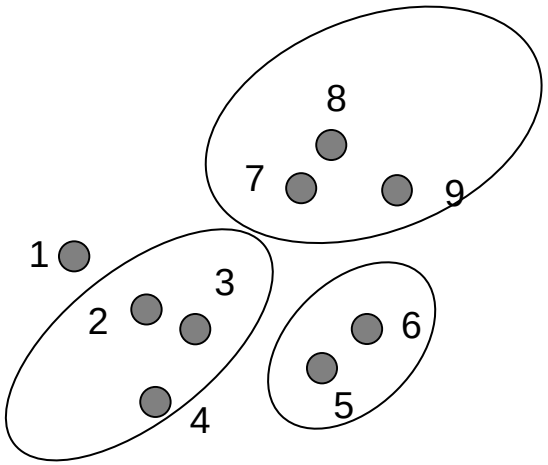


- Reprezentarea unei dendrograme: ca set de triplete ordonate

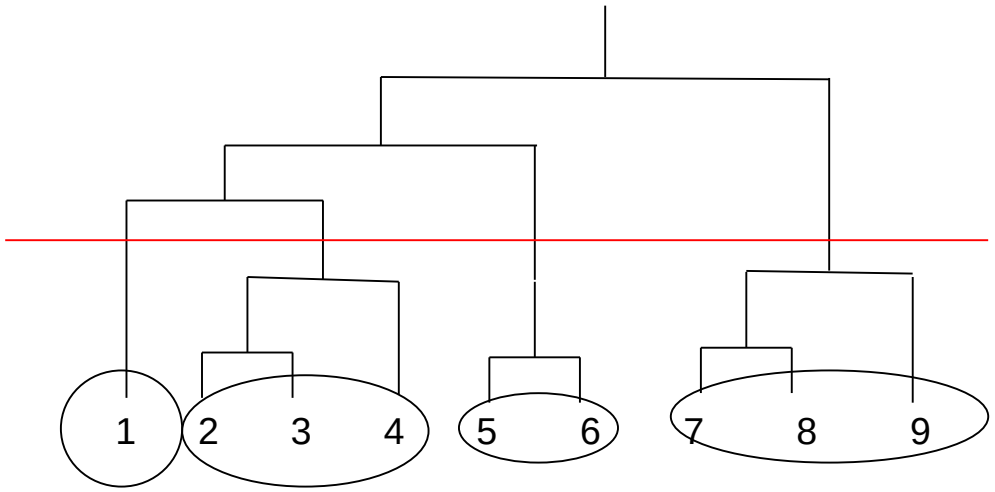
(nivel, nr de clustere, clustere)

{(0,9,{{1},{2},…,{9}}) ,(1,6,{{1},{2,3},{4},{5,6},{7 ,8},{9}}),
(2,4,{{1},{2,3,4},{5,6},{7,8,9}}), (3,3,{{1,2,3,4},{{5,6},{7,8,9}}),
(4,2,{{1,2,3,4,5,6},{7,8,9}),(5,1,{{1,2,3,4,5,6,7,8,9}})}

Metoda aglomerativă



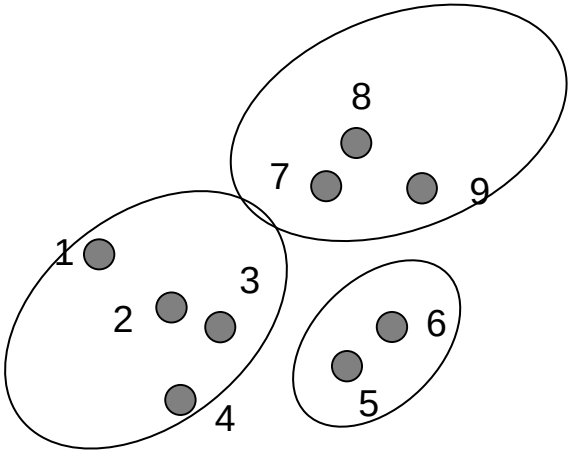
- Pentru a obţine o partiţie dendrograma trebuie secţionată



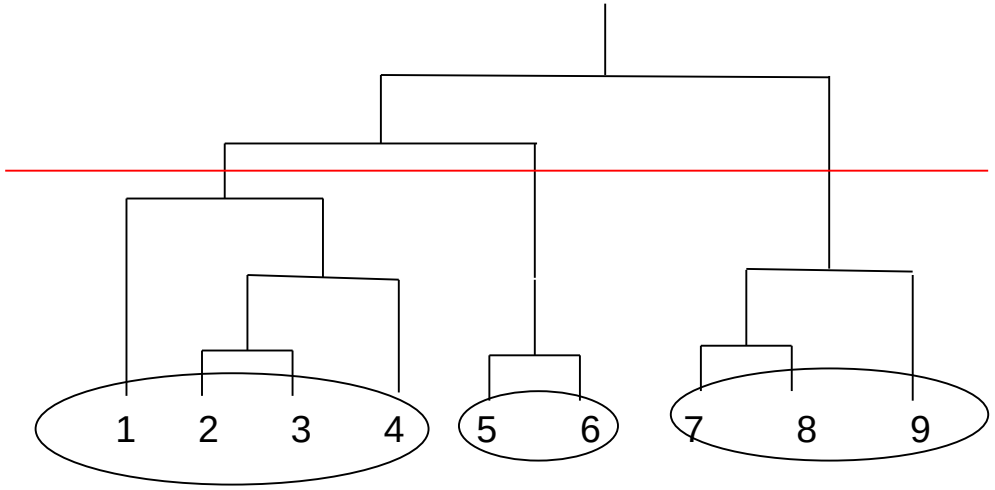
Partiţie:

C1={1}
C2={2,3,4}
C3={5,6}
C4={7,8,9}

Metoda aglomerativă



- Schimbând nivelul de secţionare se obţine o altă partiţie



Partition:

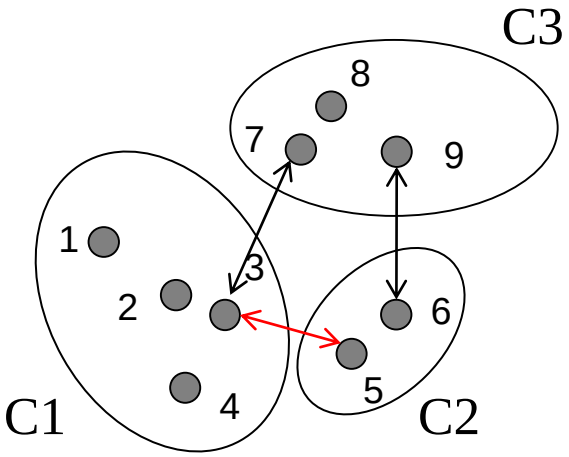
C1={1,2,3,4}
C2={5,6}
C3={7,8,9}

Metoda aglomerativă

Problema: care e criteriul de selecţie a clusterelor care se reunesc?

Răspuns: se foloseşte o măsură de disimilaritate între clustere; sunt mai multe moduri de a calcula această măsură:

- Single-linkage: cea mai mică disimilaritate (distanţă) între datele aparţinând unor clustere diferite



$$D_{SL}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

Obs:

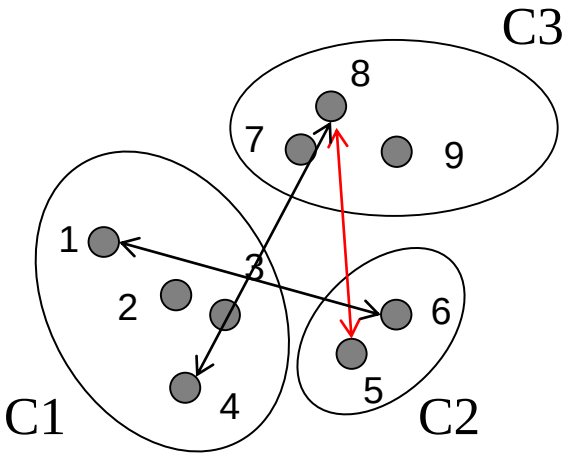
- Single link permite identificarea unor clustere care nu au neapărat formă elipsoidala
- Este sensibil la erori si date atipice

Metoda aglomerativă

Problema: care e criteriul de selecţie a clusterelor care se reunesc?

Răspuns: se foloseşte o măsură de disimilaritate între clustere; sunt mai multe moduri de a calcula această măsură

- Complete-linkage: cea mai mare disimilaritate (distanţă) între datele aparţinând unor clustere diferite



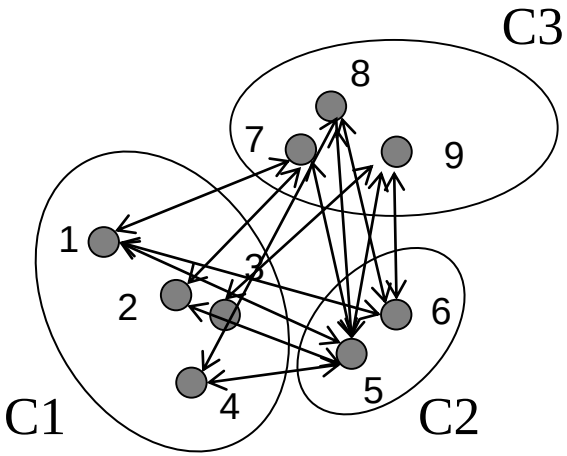
$$D_{CL}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

Metoda aglomerativă

Problema: care e criteriul de selecţie a clusterelor care se reunesc?

Răspuns: se foloseşte o măsură de disimilaritate între clustere; sunt mai multe moduri de a calcula această măsură

- Average-linkage: media distanţelor dintre datele aparţinând unor clustere diferite



$$D_{AL}(C_i, C_j) = \frac{1}{card(C_i)card(C_j)} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Metoda aglomerativă

Problema: care e criteriul de selecţie a clusterelor care se reunesc?

Răspuns: se foloseşte o măsură de disimilaritate între clustere; sunt mai multe moduri de a calcula această măsură

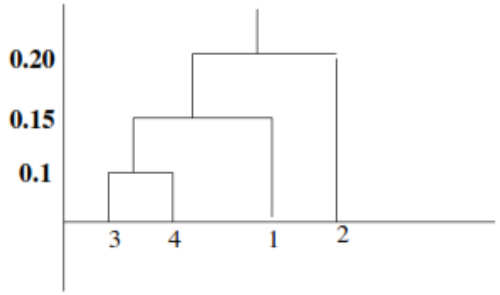
- Criteriul Ward: se alege perechea de clustere a căror reuniune conduce la cea mai mică creștere în varianța intra-cluster (SSE(X,C))

Metoda aglomerativă

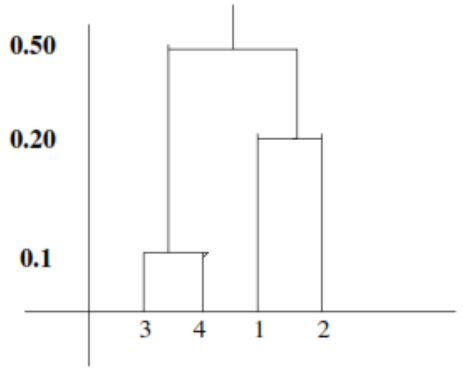
Măsura de disimilaritate folosită influenţează rezultatul grupării:

	1	2	3	4
1	0.0	0.20	0.15	0.30
2	0.20	0.0	0.40	0.50
3	0.15	0.40	0.0	0.10
4	0.30	0.50	0.10	0.0

(a) Dissimilarity Matrix



(b) Single Link



(c) Complete Link

[Data Clustering: Algorithms and Applications, 2014]

Metoda aglomerativă

Algoritm

Input : set de date cu N instanţe

$X=\{x_1,x_2,\ldots,x_N\}$ + matrice disimilaritate D

Output: dendrograma (set de triplete)

```
agglomerative(X,D)
level=0;  k=N
C={{x1},{x2},...,{xN}};  DE={(level,k,C)}
repeat
  oldk=k
  level=level+1
  (k,C)=mergeClusters(k,C,D)
  D(k) =  recompute (D(oldK), C)
  DE=union (DE, (level,k,C))
  k = k-1
until k=1
```

Obs

- Funcţia mergeClusters identifică cele mai apropiate clustere şi le reuneşte
- Funcția recompute recalculează matricea de similaritate corespunzătoare setului curent de clustere (folosind single/complete/average linkage)
- Algoritmul are complexitate pătratică în raport cu numărul de date din set ($O(N^2)$)
- Este senzitiv la erorile din date

Metoda divizivă

Structura generică

Input : set de date cu N instanţe $X=\{x_1,x_2,\ldots,x_N\}$

Output: dendrograma (tree) T

divisive(X,D)

Initializează arborele T cu nodul rădăcină (corespunde întregului set de date)

Repeat

Selectează un nod frunză L din T (based on a specific criterion)

Utilizează un algoritm partiţional de clustering pentru a descompune L in $L_1,L_2,\ldots L_k$

Adaugă $L_1,L_2,\ldots L_k$ ca fii ai lui L în arborele T

until <a stopping criterion>

Obs: algoritmul partiţional poate fi kMeans; un caz particular este bisecting kMeans care se bazează pe partiţionarea unui cluster în două alte clustere (aplicând kMeans pt k=2)

Bisecting Kmeans

- Varianta de algoritm de bisecţie bazat pe Kmeans

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

Eficientizarea algoritmilor ierarhici

- Algoritmii ierarhici tradiționali au ordin pătratic de compexitate (în raport cu numărul de date din set) – construirea dendrogramei devine costisitoare în cazul prrelucrării unui volum mare de date

- Variante mai eficiente:
  - BIRCH – Balanced Iterative Reducing and Clustering using Hierarchies (1996) – constă în două etape:
    - Se construiește o structura arborescentă care reprezintă o variantă comprimată a datelor inițiale
    - Se aplică o strategie de grupare pe datele corespunzătoare nodurilor frunză ale structurii arborescente construite în prima etapă

  - CURE – Clustering using Representatives (1998)
    - Fiecare cluster este specificat prin mai multi reprezentanți
    - Gruparea este constituită folosind eșantioane aleatoare din setul inițial de date

Cursul următor

- Algoritmi bazați pe analiza densității datelor
  - DBSCAN
  - DENCLUE

- Algoritmi bazați pe modele probabiliste
  - EM (Expectation Maximization)