Curs 7:

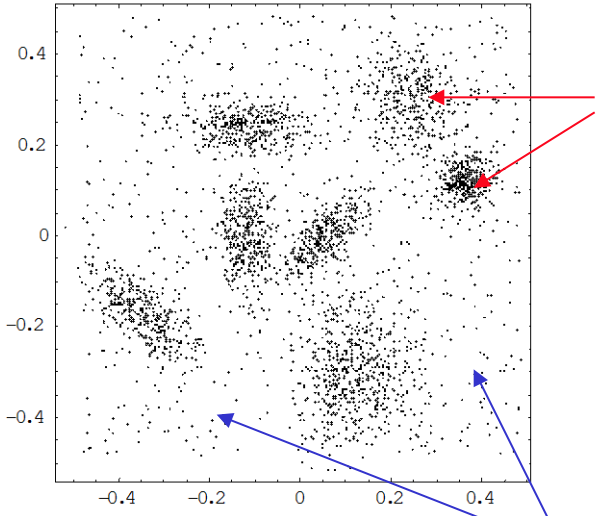
Gruparea datelor (II)

Structura

- Metode bazate pe densitate
  - DBSCAN
  - DENCLUE
- Metode probabiliste
  - EM - Expectation Maximization

Metode bazate pe densitate

- Cluster = grup dens de date similare separate de regiuni cu densitate mai mică de date
- Idee de bază: estimarea densităţii locale a datelor
  - se determină numărul de date din vecinătatea punctului analizat (DBSCAN)
  - se utilizează funcţii de influenţă pt estimarea densităţii (DENCLUE)



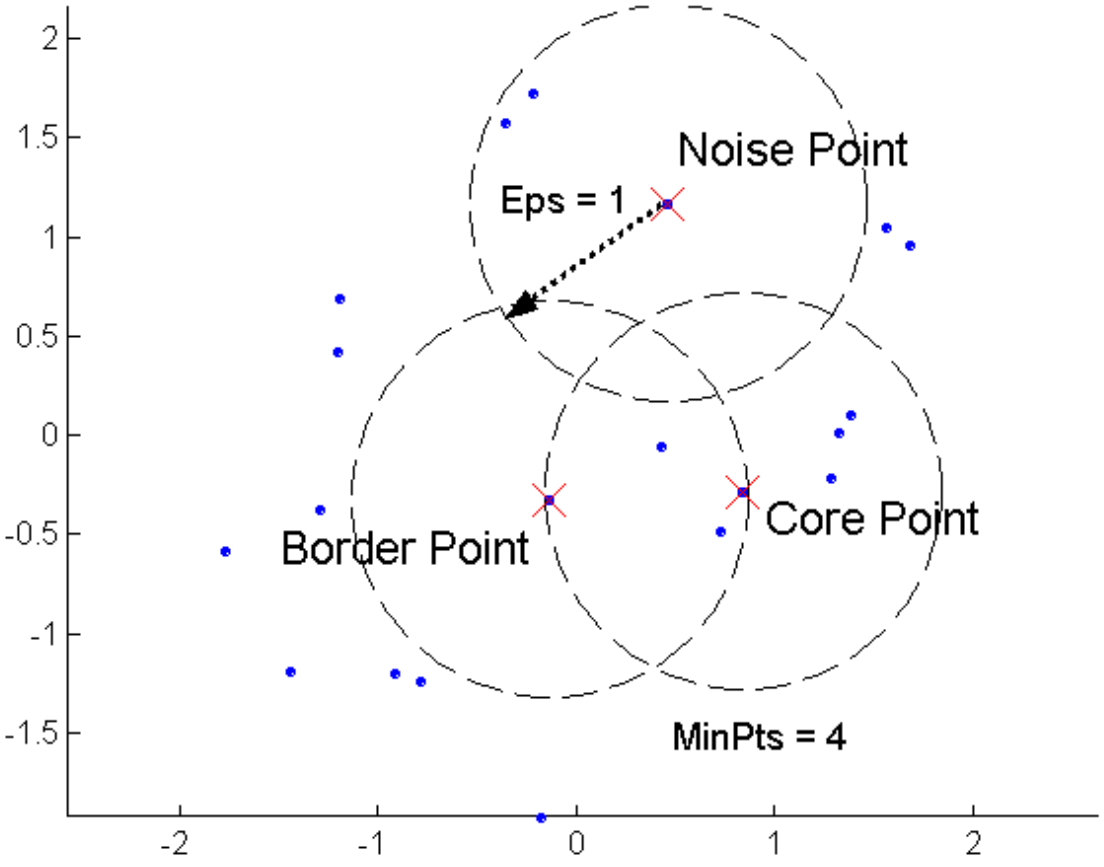
- Problema principală:
  - Cum se estimează densitatea?

DBSCAN

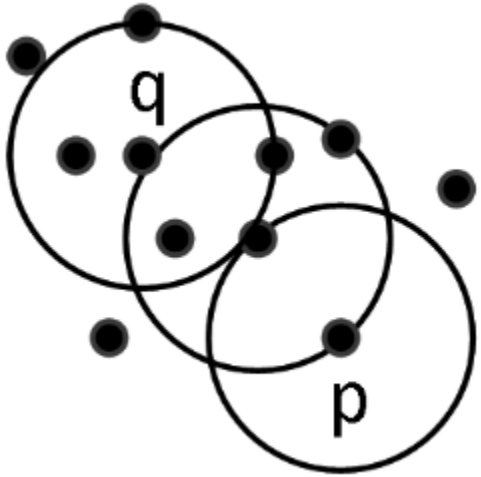
DBSCAN [M.Ester, H Kriegel et al, 1996] este un algoritm de grupare bazat pe următoarele elemente:

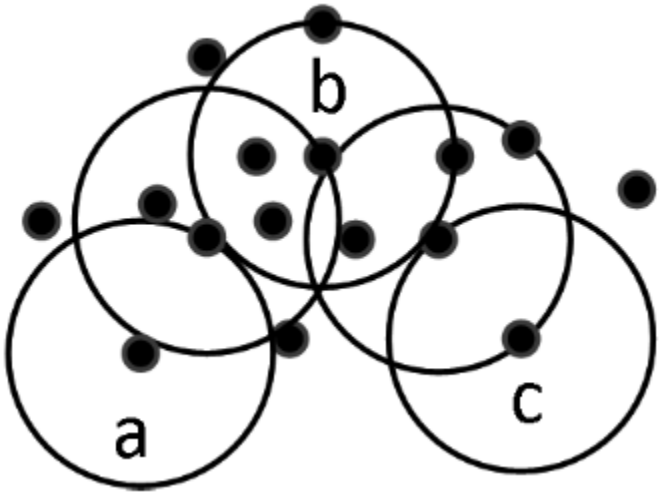
- Densitatea estimată într-un punct = numărul de puncte aflate în vecinătatea definită de o anumită rază (Eps)
- Un punct este considerat punct nucleu (core point) dacă numărul de puncte din vecinătatea sa depăşeşte un prag(MinPts); acestea sunt puncte considerate a fi în interiorul clusterului
- Un punct frontieră (border point) are un număr de vecini mai mic decât MinPts dar este în vecinătatea unui punct nucleu; Două puncte sunt considerate conectate dacă unul este în vecinătatea celuilalt
- Un punct q est accesibil direct (directly density reachable) dintr-un punct nucleu p dacă e în vecinătatea lui p; accesibilitatea în sens general este definită ca fiind închiderea tranzitivă a relaţiei de accesibilitate directă (există un lanţ de puncte nucleu ce începe cu p şi se termină cu q cu proprietatea că fiecare e direct accesibil dintr-un punct anterior)
- Un punct de tip zgomot (noise point) este orice punct care nu e nici nucleu nici frontieră

DBSCAN


Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2

DBSCAN





p este accesibil din q

a este accesibil din b
c este accesibil din b
=> a şi c sunt conectate

Obs:

- Două pcte, a şi b, dacă există un punct c astfel încât c este accesibil atât din a cât şi din b
- Două puncte conectate ar trebui să aparţină aceluiaşi cluster => un cluster definit pe baza densităţii este un set maximal de date conectate

DBSCAN

```
DBSCAN (SetOfPoints, Eps, MinPts)      // SetOfPoints is UNCLASSIFIED
    ClusterId := nextId(NOISE);
    FOR i FROM 1 TO SetOfPoints.size DO
        Point := SetOfPoints.get(i);
        IF Point.ClusterId = UNCLASSIFIED THEN
            IF ExpandCluster(SetOfPoints, Point,ClusterId, Eps, MinPts)
              THEN  ClusterId := nextId(ClusterId)
            END IF
        END IF
    END FOR
END; // DBSCAN
```

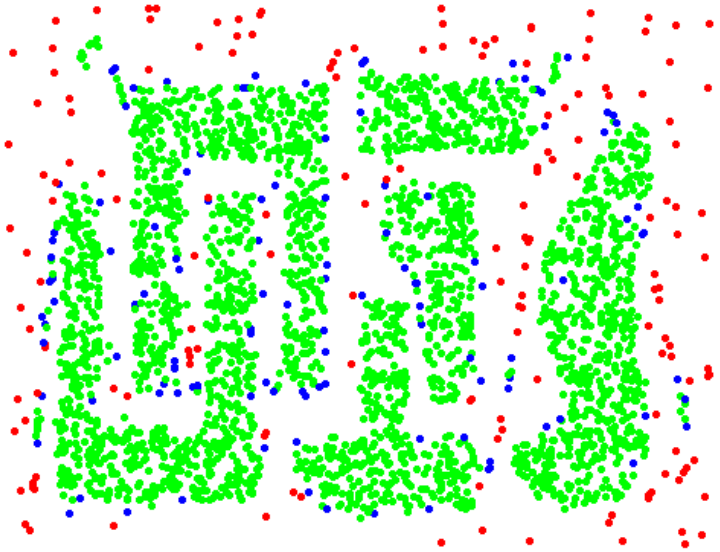
[M.Ester, H Kriegel et al, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 1996]

DBSCAN

```
ExpandCluster(SetOfPoints, Point, ClId, Eps, MinPts) : Boolean;
   seeds:=SetOfPoints.regionQuery(Point,Eps);
   IF seeds.size<MinPts THEN SetOfPoint.changeClId(Point,NOISE); RETURN False;
  ELSE // all points in seeds are density-reachable from Point
       SetOfPoints.changeClIds(seeds,ClId); seeds.delete(Point);
       WHILE seeds <> Empty DO
          currentP := seeds.first();    result := SetOfPoints.regionQuery(currentP, Eps);
          IF result.size >= MinPts THEN
            FOR i FROM 1 TO result.size DO  resultP := result.get(i);
                  IF resultP.ClId IN {UNCLASSIFIED, NOISE} THEN
                       IF resultP.ClId = UNCLASSIFIED THEN seeds.append(resultP); END IF;
                       SetOfPoints.changeClId(resultP,ClId);
                 END IF; // UNCLASSIFIED or NOISE
           END FOR;  END IF; // result.size >= MinPts
         seeds.delete(currentP); END WHILE; // seeds <> Empty
       RETURN True;  END IF;   END;
```

DBSCAN

Date (puncte) de prelucrat

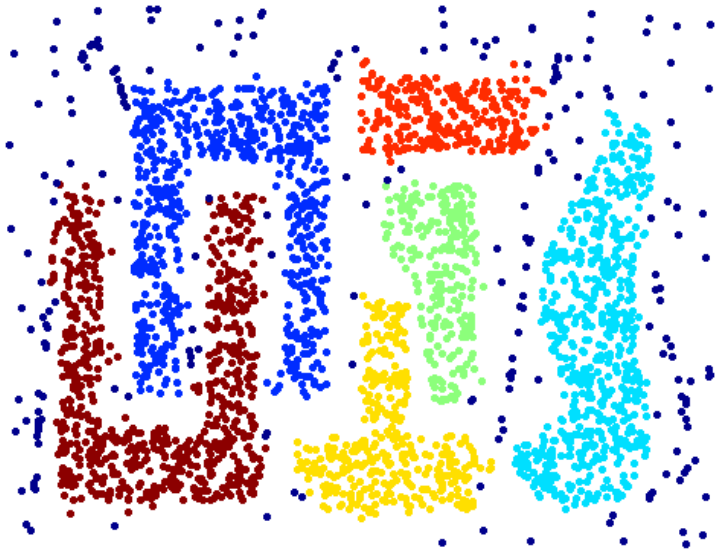


Tipuri de puncte : core, border şi noise

Eps = 10, MinPts = 4

DBSCAN

Date

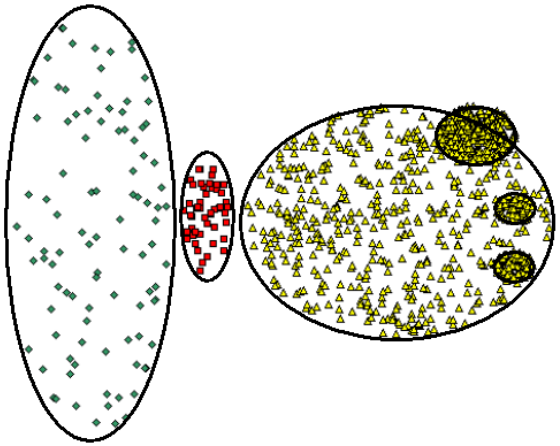


Clustere

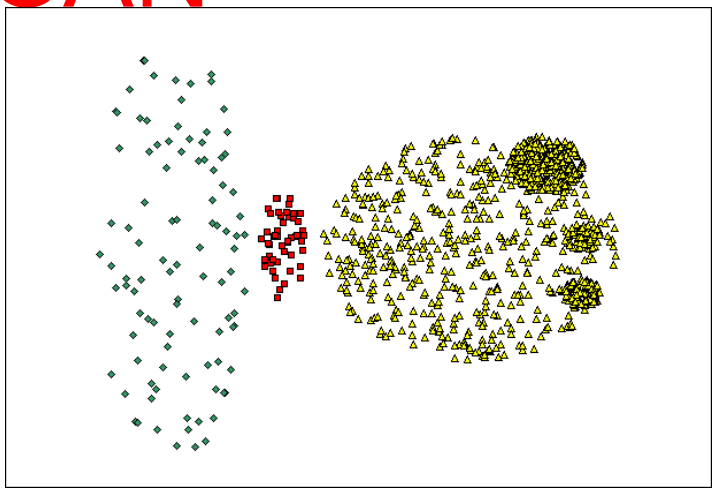
Specific:

- Permite identificarea clusterelor de diferite forme
- Robust

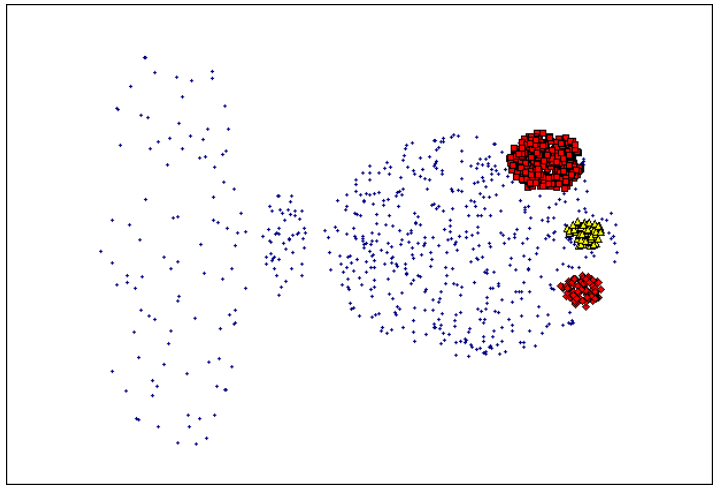
DBSCAN



Original Points



(MinPts=4, Eps=9.75).



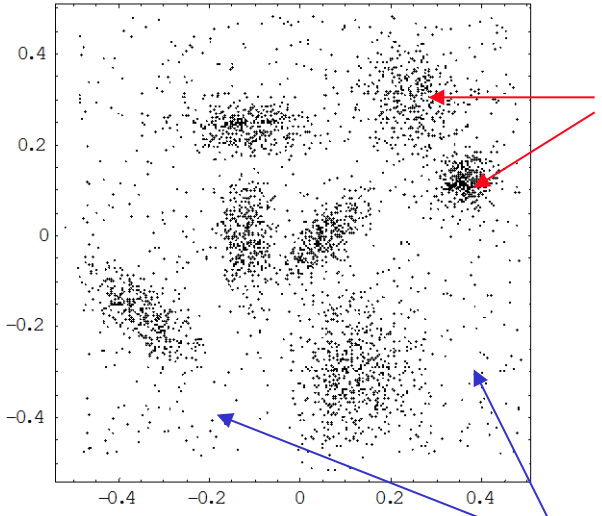
Nu este adecvat pt:

- variaţii în densitatea datelor
- date de dimensiuni mari (cu multe atribute)

DENCLUE

- Cluster = grup dens de date similare separate de regiuni cu densitate mai mică de date

- Idee de bază: estimarea densităţii locale a datelor
  - se utilizează funcţii de influenţă pt estimarea densităţii



Funcţie de influenţă

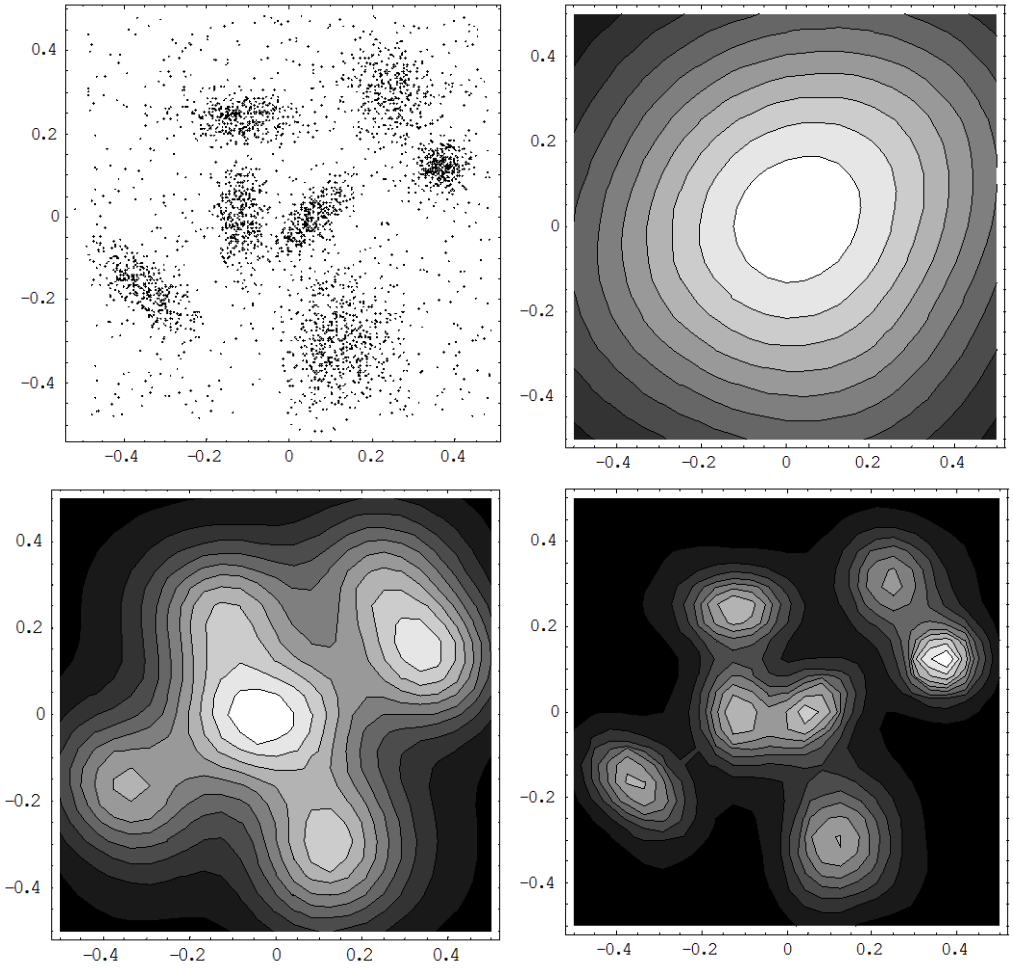
$$I_y(x) = \frac{1}{\sigma^{n/2}} \exp\left(-\frac{\sum_{j=1}^{n} (x_j - y_j)^2}{2\sigma^2} \right)$$

Funcţie de densitate

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} I_{x_i}(x)$$

DENCLUE

- Forma funcţiei de densitate depinde de valoarea lui σ
- Dacă valoarea lui σ este adecvată, maximele locale ale funcţiei de densitate corespund reprezentanţilor clusterilor
- Pt valori mari ale lui σ fctia de densitate are un maxim unic
- Pt valori prea mici ale lui σ maximele locale corespund unor vârfuri izolate şi pot fi dificil de detectat



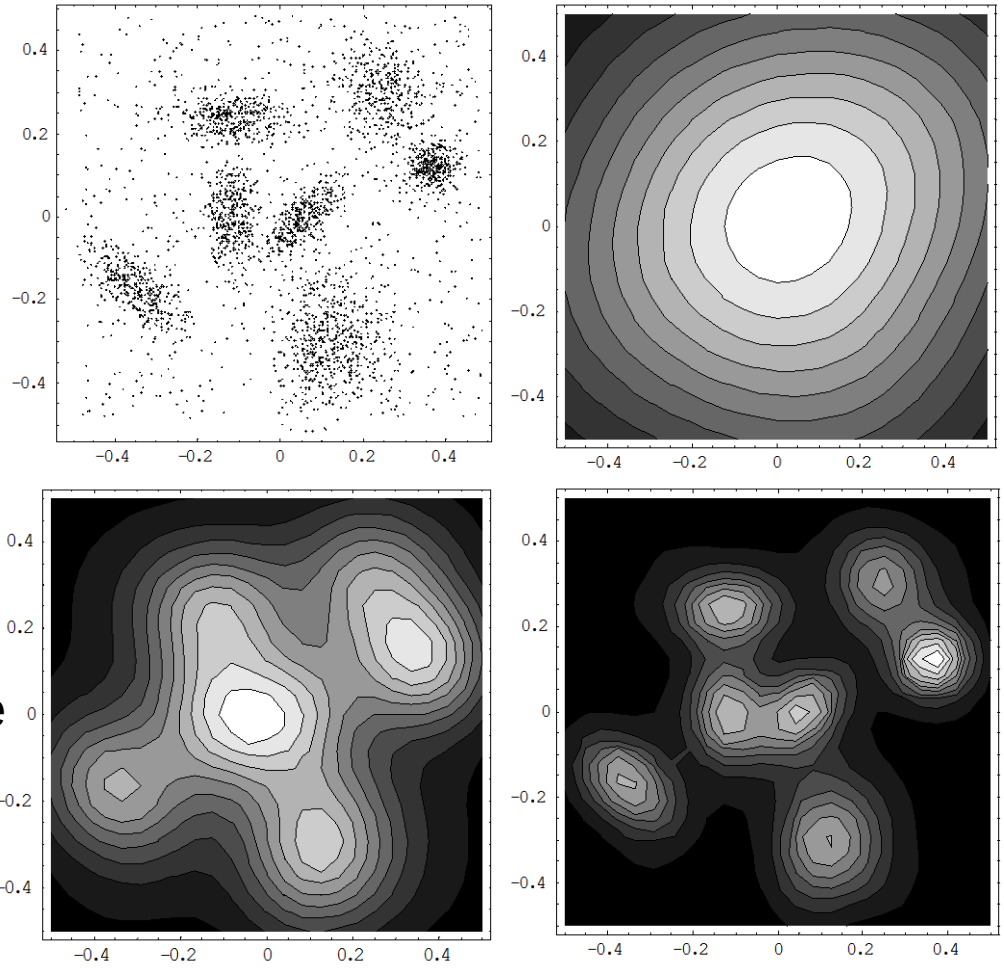
σ=0.1 σ=0.05

DENCLUE

Ideea algoritmului DENCLUE [Hinneburg, Keim – 1998]: se aplică de căutare de tip gradient pornind de la punctele din setul de date cu scopul identificării maximelor locale

Variante:

- Fiecare maxim local corespunde unui cluster (clusterele detectate vor fi sferice sau elipsoidale)
- Un cluster corespunde unui set de maxime locale “învecinate” (se pot identifica clustere de forma arbitrara)



σ=0.1 σ=0.05

Metode probabiliste

Idee de bay[:

- Datele sunt generate de un proces stohastic (o mixtură de distribuţii de probabilitate, fiecare dintre ele corespunzând unui cluster)
- Scopul algoritmului de grupare este de a descoperi modelul probabilist, adică de a identifica distribuţiile de probabilitate

Exemple:

- Algoritmul Expectation–Maximization (EM) – se bazează pe următoarele ipoteze:
  - Fiecare dată este generată de o distribuţie de probabilitate (tipul de distribuţie depinde de natura datelor)
  - În procesul de generare a datelor fiecare dintre distribuţiile de probabilitate este selectată la rândul ei cu o anumită probabilitate

Algoritmul EM

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

Iniţializarea parametrilor modelelor şi a probabilităţilor de selecţie

- (E-Step) Se determină probabilitatea de asignare a fiecărei date la clustere (folosind valorile curente ale parametrilor)
- (M-Step) Se determină parametrii modelului folosind valorile curente ale probabilităţilor de asignare a datelor la clustere

Obs:

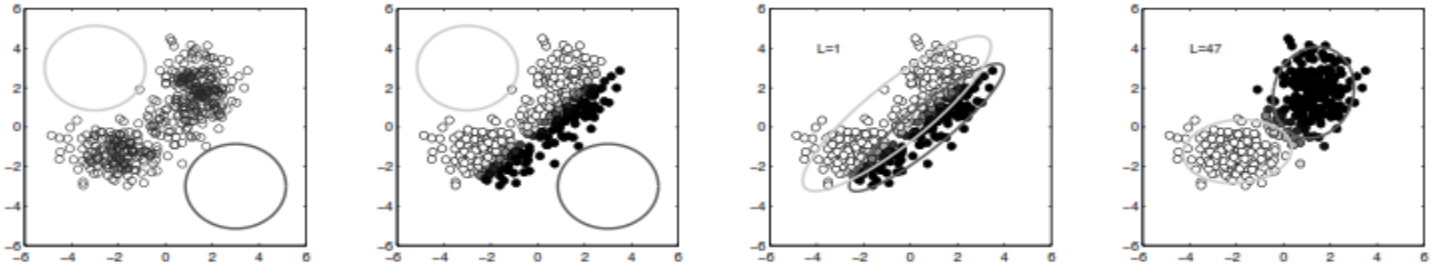
- In cazul datelor numerice se poate folosi distribuţia normală multi-dimensională pentru a modela datele aparţinând unui cluster (modelul de generare a datelor este o mixtură de gaussiene). Parametrii care se estimează sunt media şi matricea de covarianţă.

EM algorithm

Algorithm 11 EM for Gaussian Mixtures

Given a set of data points and a Gaussian mixture model, the goal is to maximize the log-likelihood with respect to the parameters.

1: Initialize the means μ_k^0, covariances Σ_k^0, and mixing probabilities π_k^0.
2: **E-step**: Calculate the responsibilities $\gamma(z_{nk})$ using the current parameters based on Equation (3.13).
3: **M-step**: Update the parameters using the current responsibilities. Note that we first update the new means using (3.12), then use these new values to calculate the covariances using (3.14), and finally reestimate the mixing probabilities using (3.15).
4: Compute the log-likelihood using (3.10) and check for convergence of the algorithm. If the convergence criterion is not satisfied, then repeat steps 2–4; otherwise, return the final parameters.



Data Clustering Algorithms and applications (ed. CC Aggarwal, CK Reddy, 2014)

Algoritmul EM

Ecuaţii care intervin in algoritmul EM

$$p(\mathbf{x}_n|\Theta) = p(\mathbf{x}_n|\pi,\mu,\Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k). \qquad (3.10)$$

$$\mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\}$$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \qquad (3.12)$$

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j,\Sigma_j)}. \qquad (3.13)$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}. \qquad (3.14)$$

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N} \qquad (3.15)$$

Data Clustering Algorithms and applications (ed. CC Aggarwal, CK Reddy, 2014)