Tehnica *k-means* (k-medii) pentru probleme de grupare

Gruparea datelor reprezintă unele dintre sarcinile de bază din arii de procesare a datelor precum data mining[1] și învățarea automată (Hastie et al. 2009), (Haykin 2009). Problema grupării datelor este una de reprezentativitate, în sensul că toate instanțele care aparțin unui grup trebuie, într-un sens optimal, să fie cel mai bine reprezentate/delimitate spațial (din punct de vedere al unei metrici) de acel grup caracterizat în mod unic de centrul grupului respectiv. Se poate considera că tehnicile de grupare a datelor reprezintă precursori ai tehnicilor de clasificare, devreme ce etichetarea datelor pentru a rezolva ulterior probleme de clasificare este un proces deosebit de dificil pentru seturi de date foarte mari în care etichetarea manuală reprezintă o opțiune costisitoare. Pe de altă parte, tehnicile de clasificare au ca obiectiv învățarea unor relații între atributele de intrare și etichetele de ieșire, relații pe baza cărora noi instanțe care apar pot fi clasificate. În timp ce tehnicile de grupare presupun disponibilitatea tuturor datelor pentru faza de grupare.

Procesul de grupare a datelor, este, prin natura sa, un proces dificil de automatizat, ce poate fi cel mai bine acceptat ca un proces iterativ care implică interactivitate cu utilizatorul: pentru alegerea parametrilor tehnicilor de grupare, pentru preprocesarea datelor, dar și pentru verificarea soluțiilor produse de aceste tehnici, verificare care necesită un proces aprioric conștient din partea utilizatorului. Cu alte cuvinte, având mulțimi de date la dispoziție, prin intermediul tehnicilor de vizualizare, de conștientizare a atributelor instanțelor de date, utilizatorul intuiește de la bun început sensul în care datele pot fi grupate și definește în mod corespunzător metrici, inițializări de parametri cum ar fi număr de grupuri, centrele inițiale, etc. La final unui proces de grupare, soluțiile pot fi vizualizate pentru a confirma dacă sunt „bune” sau necesită „îmbunătățiri”.

Există mai mult moduri de a defini grupurile, ceea ce conduce la mai multe modele de grupare și, corespunzător, la diverși algoritmi:

- modele de grupare ierarhică de agregare (fiecare instanță începe ce un grup de sine stătător și aceste grupuri sunt mai apoi agregate) și de divizare (în care toate instanțele formează inițial un singur grup care va fi mai apoi subdivizat în alte grupuri), cu algoritmii HAC (engl. *Hierarchical Agglomerative Clustering*) și respectiv DIANA (engl. *DIvisive ANAlysis Clustering*),

- modele de tip median în care fiecare grup cu instațele aferente sunt reprezentate de o valoare corelată cu media atributelor instanțelor, algoritmul reprezentativ fiind aici k-means,

- modele de tip distribuție statistică, cu algoritmul reprezentativ Expectation-Maximization,

- modele bazate pe densitatea instanțelor, cu algoritmul DBSCAN (engl. *Density-based spatial clustering of applications with noise*) reprezentativ,

- modele bazate pe reprezentare de tip grafuri, cu algoritmul HCS (engl. *Highly Connected Subgraphs*),

- modele de tip rețea neuronală hartă cu auto-organizare (hartă/rețea Kohonen).

În cadrul acestui capitol vor fi studiate două tehnici de grupare populare: tehnica k-means și tehnica Expectation-Maximization.

În particular, tehnica *k-means* (cu algoritmul *k-means* corespunzător) este una dintre cele mai simple și utilizate tehnici de grupare a datelor ce aparțin paradigmei de învățare nesupervizată. Aplicațiile tehnicii sunt diverse și variază de la gruparea textelor (Andrews and Fox, 2007; Dhillon et al., 2003),

[1] Data mining cuprinde toate metodele și tehnicile folosite pentru a extrage cunoștințe din date.

prelucrarea imaginilor (Philibin et al., 2007; Shi and Malik, 2000), analiza datelor genetice (Baldi and Hatfield, 2002; Lukashin et al., 2003) la prelucrarea datelor climaterice (Steinbach et al., 2003).

Fie un set de instanțe *d*-dimensionale (cu *d* atribute numerice) măsurate de forma $D = \{\mathbf{x}_1,...,\mathbf{x}_n\}, \mathbf{x}_i \in \Re^d$. Problema k-means este aceea de a atribui cele *n* instanțe de date unor *K* grupuri (sau clustere) C_i caracterizate de centrele din mulțimea $C = \{\mathbf{c}_i \in \Re^d, i = \overline{1,K}\}$. Problema de optimizare specifică k-means este formulată ca în relația

$$\mathbf{c}_i^* = \arg\min_{\mathbf{c}_i} J(\mathbf{c}_i) = \sum_{j=1}^{K} \sum_{\mathbf{x}_i \in C_j} \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2 . \quad (1)$$

Problema k-means este una de tip NP-hard[2]. O soluție iterativă este redată prin pașii algoritmul k-means de mai jos:

1. Sunt selectate arbitrar (sau folosind tehnici de inițializare specifice) cele *K* centre ale grupurilor, $C = \{\mathbf{c}_i \in \Re^d, i = \overline{1,K}\}$.
2. Pasul de atribuire: fiecare grup C_i va conține acele instanțe din *D* care sunt mai apropiate de centrul $\mathbf{c}_i$ decât față de oricare alt centru $\mathbf{c}_j, j \neq i$.
3. Pasul de recalculare a centrelor fiecărui grup: pentru fiecare grup vor fi recalculate centrele de cu formula $\mathbf{c}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$, unde $|C_i|$ reprezintă numărul elementelor din mulțimea C_i.
4. Vor fi reluați pașii 2 și 3 până când centrele nu se vor mai schimba.

Algoritmul k-means anterior este unul iterativ care efectuează îmbunătățiri la fiecare pas și converge la o soluție local optimală. În multe situații practice, este recomandată reinițializarea aleatoare a centrelor și reluarea pașilor algoritmului, pentru a găsi soluții mai bune. Atunci când datele pot fi ușor explorate vizual și gruparea instanțelor este evidentă concomitent cu evidența numărului grupurilor, inițializările pot fi efectuate manual. Totuși, chiar și în aceste situații în care instanțele sunt în mod natural grupate și inițializarea centrelor se face manual, algoritmul k-means poate conduce la soluții de grupare nenaturale. Cel mai important parametru este numărul grupurilor, care este dificil însă de ales fără o analiză apriorică a datelor.

Există multe soluții de inițializare a centrelor grupurilor. Cea mai populară pare a fi soluția k-means++ cu algoritmul aferent (Arthur, Vassilivitskii; 2007). Fie *d*(**x**) cea mai scurtă distanță de la o instanță **x** la cel mai apropiat centru din lista centrelor deja selectate. Atunci, k-means++ presupune:

1. Alegerea primului centru $\mathbf{c}_1$, în mod aleator, dintre instanțele din mulțimea *D*.
2. Se alege un nou centru $\mathbf{c}_i$ dintre instanțele rămase în $\mathbf{x} \in D$, cu probabilitatea $\frac{d(\mathbf{x})^2}{\sum_{\mathbf{x} \in D} d(\mathbf{x})^2}$.

   Factorul de la numitor are rol de normalizare. Pasul presupune ca pentru fiecare instanță rămasă în *D* care nu este deja centru, să fie cunoscută distanța până la cel mai apropiat centru din lista centrelor deja selectate (care presupune calculul tuturor distanțelor până la centrele deja selectate și reținerea cele mai scurte dintre aceste dinstanțe). Instanța cea mai îndepărtată de centrele deja selectate va avea cea mai mare probabilitate de a fi selectată ca și un nou centru.
3. Repetarea pasului 2 până când au fost selectate *K* centre.

[2] Clasa problemelor NP este reprezentată de problemele de decizie care, în cel mai rău caz, sunt soluționate în timp polinomial. Clasa NP-hard cuprinde problemele cel puțin la fel de grele ca cele mai grele probleme din clasa NP.

După inițializare, algoritmul k-means poate fi rulat în continuare, necesitând în mod uzual un număr mult mai mic de iterații până la convergență și funcții de cost (FC) de valori mai mici, mai apropiate de minimul global. Implementarea k-means++ sugerează memorarea unor liste separate pentru mulțimea crescândă a centrelor și pentru mulțimea descrescătoare a instanțelor din *D* care devin centre.

Odată ce gruparea este efectuată și sunt cunoscute centrele grupurilor finale, soluția poate fi verificată prin reprezentări grafice, atunci când este cazul. În cazul reprezentării plane, așa-numitele diagrame Voronoi pot fi construite pentru a colora diferit regiunile din spațiu aferente punctelor celor mai apropiate de centrul fiecărui grup. În fine, soluția de grupare poate fi ușor extinsă într-una de clasificare, prin atribuirea oricărei noi instanțe la unul din grupurile obținute, pe baza criteriului de distanță.

Nu întotdeauna criteriul de grupare este cel de distanță euclideană. Pot fi utilizate alte metrici precum distanța L_1 (cityblock), distanța cosinus, distanța corelație, sau distanța hamming (pentru date binare).

În fine, algoritmul k-means se comportă cel mai bine atunci când datele prezintă tendințe de grupare relativ sferică și numărul grupurilor algoritmului este egal cu cel al grupurilor potențial evidențiate apriori.

Coeficientul de siluetă este un indicator specific al calității procesului de grupare și pentru fiecare instanță $\mathbf{x}_i$ se măsoară cu ajutorul formulei $S_i = (b_i - a_i)/\max(a_i, b_i)$, unde a_i este distanța medie de la instanța $\mathbf{x}_i$ la celelalte puncte din același cluster din care face parte $\mathbf{x}_i$, iar b_i este cea mai mică distanță medie de la $\mathbf{x}_i$ la instanțele din celelalte grupuri. Coeficientul de siluetă ia valori în intervalul $S_i \in [-1;1]$. Un coeficient S_i de valoare ridicată indică faptul că instanța $\mathbf{x}_i$ a fost bine alocată grupului din care face parte și că nu ar trebui să facă parte din alt grup. Dacă majoritatea instanțelor au un coeficient de siluetă de valoare ridicată, gruparea este “corect efectuată”. Dacă în schimb multe instanțe au coeficient de valoare mică (inclusiv negativă), înseamnă că există prea multe sau prea puține grupuri. Coeficientul de siluetă poate fi utilizat în combinație cu orice metrică de distanță.

Tehnica Expectation-Maximization pentru probleme de grupare

Algoritmul Expectation-Maximization (EM) stă la baza unei tehnici de grupare (clasificare) aparținând paradigmei învățării nesupervizate și reprezintă, într-un sens, o generalizare a algoritmului k-means prin aceea că o instanță din setul de date de grupat nu mai aparține doar unui singur grup (apartenență “hard”) ci aparține, cu diverse grade de probabilitate, tuturor grupurilor (apartenență de tip ”soft”). O altă caracteristică a algoritmului EM o reprezintă capacitatea de grupare în condițiile unor instanțe incomplete în care o parte din atribute sunt nemăsurabile (se mai numesc și atribute ascunse). Tehnica EM este la rândul ei una populară, cu aplicații diverse precum analiza datelor genetice (Slatkin and Excoffer, 1996; Fallin and Schork, 2000), învățarea prein recompensă (Dayan and Hinton, 2006), segmentarea imaginilor (Ramme et al., 2009; Zhang et al., 2001; Fatakdawala et al., 2010; Zafrir et al., 2012), identificarea sistemelor dinamice cu parametri variabili în timp (Frenkel and Feder, 2009), învățarea topologiilor rețelelor de senzori (Marinakis et al., 2005).

Pentru a ilustra și motiva algoritmul EM, se consideră o problemă de estimare a parametrilor unui model de tip Gaussiene combinate (engl. Gaussian Mixed Model (GMM)). Se presupune problema grupării (/clasificării) unor autovehicule în comerciale și personale, folosind funcția de densitate de probabilitate (f.d.p.) construită pe baza masei autovehiculului ilustrată în Fig. 1. Deoarece f.d.p. este multimodală, este foarte probabil ca să existe un atribut ascuns dar nemăsurat, în cazul de față tipul de autovehicul (sedan, compactă, van, de teren, microbuz). Pentru un tip fixat de autovehicul, masa este normal distribuită (Gaussiană), astfel că pentru toate tipurile posibile, funcția de densitate va fi cel mai bine reprezentată de o combinație de f.d.p. normale.

În continuare este redat un cadru teoretic mai larg privind algoritmul EM pentru modelele GMM, pentru probleme de grupare (Poczos 2014). Fie un set de date *d*-dimensionale (cu *d* atribute) măsurate de forma $D = \{\mathbf{x}_1,...,\mathbf{x}_n\}, \mathbf{x}_i \in \Re^d$, unde $\mathbf{X} \in \Re^d$ este o variabilă aleatoare și $\mathbf{x}_i \in \Re^d$ este o realizare a sa (o măsurare). Un model de distribuție Gaussiană multi-modală multi-variabilă (d-variabilă în acest caz) care combină *K* distribuții normale se bazează pe faptul că fiecare instanță care aparține acestei distribuții provine dintr-una din cele *K* distribuții normale $\mathrm{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), j = \overline{1,K}$ caracterizate de media $\boldsymbol{\mu}_j = E[\mathbf{X}] \in \Re^d$ și de covarianța $\boldsymbol{\Sigma}_j(\mathbf{X}) = E[(\mathbf{X}-\boldsymbol{\mu}_j)^T(\mathbf{X}-\boldsymbol{\mu}_j)] \in \Re^{d\times d}$, statistici calculate în raport cu variabila aleatoare **X** (vezi și Fig. 2). Funcția de densitate de probabilitate (f.d.p.) a unei astfel de distribuții normale multivariabilă poate fi modelată de expresia[3]

$$fdp(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{d/2} \mid \boldsymbol{\Sigma}_j \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right). \quad (2)$$

Fie atributul ascuns y_i reprezentând indexul celei mai probabile distribuții care a generat o anumită instanță $\mathbf{x}_i$ și fie probabilitatea de a genera o anumită instanță $\mathbf{x}_i$ din modelul GMM care să aparțină distribuției (clusterului sau grupului) $j = \overline{1,K}$ notată cu $\pi_j = P(y_i = j)$. Suma tuturor π_j trebuie să fie egală cu 1 (o instanță provine cu certitudine din una din distribuții) și trebuie acceptat și faptul că o anumite valoare exactă de apariție a variabilei aleatoare "activează" mai multe f.d.p. simultan. Deoarece o variabilă aleatoare care *provine și este extrasă* din distribuția *j* va avea probabilitatea de apariție dată de f.d.p normală[4]

$$P(\mathbf{X} = \mathbf{x} \mid y = j) = fdp(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \quad (3)$$

rezultă că probabilitatea de apariție a instanței **x** aparținând modelului GMM depinde de probabilitățile condiționate de apartenența la distribuțiile normale $j = \overline{1,K}$ prin relația

$$P(\mathbf{X} = \mathbf{x}) = \sum_{j=1}^{K} P(\mathbf{X} = \mathbf{x} \mid y = j)P(y = j) = \sum_{j=1}^{K} fdp(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\pi_j, \quad (4)$$

coeficienții π_j având aici rolul de ponderi în combinarea f.d.p. (a distribuțiilor) normale care generează modelul GMM. Poate fi considerată în continuare *mulțimea de instanțe completate* ca fiind de forma $D_c = \{(\mathbf{x}_1, y_1),...,(\mathbf{x}_n, y_n)\}$.

---

[3] De exemplu, o matrice de covarianță diagonală de forma $\boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma_{11,i}^2 & 0 \\ 0 & \sigma_{22,i}^2 \end{pmatrix}$ care corespunde la atribute independente statistic, conduce la $\mid \boldsymbol{\Sigma}_i \mid^{1/2} = \sqrt{(\sigma_{11,i}^2\sigma_{22,i}^2 - 0)} = \sigma_{11,i}\sigma_{22,i}$.

[4] În realitate, pentru o variabilă aleatoare continuă (și scalară), probabilitatea de apariție calculată cu ajutorul f.d.p. este 0, însă ceea ce poate fi calculat mai exact este probabilitatea ca o variabilă aleatoare X să ia valori într-un anumit interval, în cazul scalar, este $P(a < X < b) = \int_a^b fdp(x, \mu, \sigma)dx$. Totuși, pentru aplicații practice, putem considera că $fdp(x, \mu, \sigma)dx$ este probabilitatea ca variabila aleatoare X să ia valori într-un interval infinitezimal mic în jurul valorii *x*.

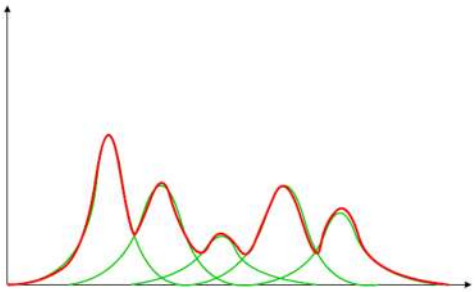

Fig. 1 F.d.p. multimodală obținută din măsurătorile maselor autovehiculelor, cu relevarea f.d.p. normale care o compun.

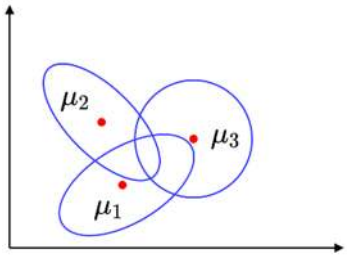

Fig. 2 Relativă la o distribuție multi-modală bi-dimensională.

Prima problemă care se pune este de a decide din care distribuție normală din cele *K* care compun modelul GMM provine o anumită instanță **x**, în cazul în care toți parametrii tuturor distribuțiilor sunt cunoscuți, adică sunt cunoscute valorile $\boldsymbol{\theta} = [\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1,...,\boldsymbol{\Sigma}_K, \pi_1,...,\pi_K]^T$ numite parametri ai GMM. Atunci, în baza Teoremei lui Bayes, probabilitatea ca instanța **x** provenind din distribuția multimodală GMM să aparțină distribuției *j* este

$$P(y=j \mid \mathbf{X}=\mathbf{x}) = \frac{P(\mathbf{X}=\mathbf{x} \mid y=j)P(y=j)}{P(\mathbf{X}=\mathbf{x})} = \frac{fdp(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\pi_j}{\sum_{i=1}^{K} fdp(\mathbf{x}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\pi_j}$$ [5], (5)

indicator care servește atribuirii instanței **x** la distribuția pentru care rezultă cea mai mare probabilitate de apartenență. Această problemă operație poate fi interpretată și ca o operație de *generare a atributelor ascunse* cunoscând anumite valori ale parametrilor modelului GMM.

Cealaltă problemă care se pune este de a determina parametrii nescunoscuți $\boldsymbol{\theta}$ care caracterizează cel mai bine distribuția care a generat datele culese *completate cu atributul ascuns y* (care poate fi calculat în baza relației (5) folosind valori apriorice ale parametrilor $\boldsymbol{\theta}$). Această problemă este una de calcul de tip estimator al verosimilității (plauzabilității) maxime (engl. *Maximum Likelihood Estimator* – (MLE)) în care parametrii $\boldsymbol{\theta}$ ai modelului GMM reprezintă soluția unui probleme de optimizare (PO) în care aceștia maximizează probabilitatea ca instanțele măsurate completate (această probabilitate este mai sus numită *funcție de verosimilitate*) să apară, condiționate de parametrii modelului GMM. Deoarece *se presupune* că instanțele din setul de date completat provin din evenimente aleatoare independente, poate fi scris că

$$P((\mathbf{X}=\mathbf{x}_1, y_1),...,(\mathbf{X}=\mathbf{x}_n, y_n) \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} P(\mathbf{X}=\mathbf{x}_i, y_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{n}\sum_{j=1}^{K} P(\mathbf{X}=\mathbf{x}_i, y_i=j \mid \boldsymbol{\theta})P(y_i=j \mid \boldsymbol{\theta}),$$ (6)

[5] Dacă probabilitățile ar fi exprimate conform $fdp(x, \mu, \sigma)dx$ pentru un interval infinitezimal dx fixat, atunci dx este factor comun la numărător și numitor și dispare din fracție prin simplificare.

care poate fi citită „*Probabilitatea ca* {prima instanță de date să ia valoarea $\mathbf{x}_1$ fiind generată din distribuția aferentă y_1 oarecare, ..., și ca ultima instanță de date să ia valoarea $\mathbf{x}_n$ fiind generată de distribuția y_n oarecare}, *condiționată de parametrii* $\boldsymbol{\theta}$*, este egală cu...*". În continuare se poate arăta că a calcula parametrii $\boldsymbol{\theta}$ care maximizează funcția de verosimilitate (5) a setului de date complet, este echivalent cu a calcula parametrii $\boldsymbol{\theta}$ care maximizează funcția de verosimilitate a setului de date incomplet, deoarece

$$P(\mathbf{X}=\mathbf{x}_1,...,\mathbf{x}_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} P(\mathbf{X}=\mathbf{x}_i \mid \boldsymbol{\theta}) \overset{(3)}{=} \prod_{i=1}^{n}\left(\sum_{j=1}^{K} \underbrace{P(\mathbf{X}=\mathbf{x}_i \mid y_i=j,\boldsymbol{\theta})P(y_i=j \mid \boldsymbol{\theta})}_{P(\mathbf{X}=\mathbf{x}_i, y_i=j \mid \boldsymbol{\theta})}\right) = \prod_{i=1}^{n}\left(\sum_{j=1}^{K} P(\mathbf{X}=\mathbf{x}_i, y_i=j \mid \boldsymbol{\theta})\right), \quad (7)$$

în ultima egalitate folosind legea probabilităților condiționate $P(A \mid B) = P(A,B)/P(B)$ care condiționate de evenimentul $\boldsymbol{\theta}$ devine $P(A \mid B,\boldsymbol{\theta}) = P(A,B \mid \boldsymbol{\theta})/P(B \mid \boldsymbol{\theta})$. Aici $P(A,B) = P(A \bigcap B)$ reprezintă probabilitatea ca evenimentele A și B să apară împreună. Deci $P(\mathbf{X}=\mathbf{x}_i \mid y_i=j,\boldsymbol{\theta}) = \dfrac{P(\mathbf{X}=\mathbf{x}_i, y_i=j \mid \boldsymbol{\theta})}{P(y_i=j \mid \boldsymbol{\theta})}$ care este folosită în (7).

Este definită astfel PO aferentă problemei de calcul al estimatorului de verosimilitate maximă sub forma

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}}\left[P(\mathbf{X}=\mathbf{x}_1,...,\mathbf{x}_n \mid \boldsymbol{\theta})\right] \overset{indep}{=} \arg\max_{\boldsymbol{\theta}}\left[\prod_{i=1}^{n}\left(\sum_{j=1}^{K} P(\mathbf{X}=\mathbf{x}_i \mid y_i=j,\boldsymbol{\theta})P(y_i=j \mid \boldsymbol{\theta})\right)\right], \quad (8)$$
$$\overset{(4)}{=} \arg\max_{\boldsymbol{\theta}}\left[\prod_{i=1}^{n}\left(\sum_{j=1}^{K} \frac{\pi_{ij}}{(2\pi)^{d/2} \mid \boldsymbol{\Sigma}_j \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_j)\right)\right)\right]$$

Funcția de cost a PO anterioare este una neliniară în raport cu argumentul, iar în practică este mai ușor de lucrat cu logaritmul FC (al funcției de verosimilitate). Relația (5) nu intervine direct în (8) însă din (5) rezultă că

$$\pi_j = P(y_i=j \mid \boldsymbol{\theta}) = \frac{P(y_i=j \mid \mathbf{X}=\mathbf{x}_i,\boldsymbol{\theta})P(\mathbf{X}=\mathbf{x}_i \mid \boldsymbol{\theta})}{P(\mathbf{X}=\mathbf{x}_i \mid y_i=j,\boldsymbol{\theta})} = \frac{R_{ij}\left(\sum_{j=1}^{K} fdp(\mathbf{x}_i,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)\pi_j\right)}{fdp(\mathbf{x}_i,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}, \quad (9)$$
$$R_{ij} = P(y_i=j \mid \mathbf{X}=\mathbf{x}_i,\boldsymbol{\theta}) = \frac{fdp(\mathbf{x}_i,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)\pi_j}{\sum_{j=1}^{K} fdp(\mathbf{x}_i,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)\pi_j},$$

în care cantitățile R_{ij} calculează pentru fiecare instanță *i* probabilitatea să fi apărut din distribuția *j*.

Ca urmare a celor două probleme expuse mai sus, se concluzionează că cele două situații aparent nesoluționabile ale problemei de grupare/clasificare în cazul în care există atribute ascunse și nu sunt cunoscuți parametrii distribuțiilor care au generat aceste date, sunt:

1) Dacă s-ar cunoaște parametrii distribuțiilor s-ar putea genera atributele ascunse folosind (5).
2) Dacă s-ar cunoaște atributele ascunse, ar putea fi calculați parametrii distribuțiilor cu (8).

Algoritmul EM oferă o soluție iterativă a situațiilor de calcul de mai sus prin care, pornind de la valori inițiale ale parametrilor necunoscuți ai distribuțiilor, alternează pașii (5) și (8) până la atingerea unei soluții local optimale. Pasul de calcul redat de (5) este pasul de **Expectation** și are rolul de a genera atributele ascunse pentru a avea disponibile date complete, pe baza cărora mai apoi pot fi calculați

parametrii distribuțiilor care compun modelul GMM, rezolvând problema (8), care reprezină pasul **Maximization** al algoritmului EM.

Pentru problema de grupare, algoritmul EM are următoarea formulare: Este cunoscut setul de date incomplete $D=\{\mathbf{x}_1,...,\mathbf{x}_n\}, \mathbf{x}_i \in \Re^d$, se cunoaște că aceste instanțe provin dintr-o distribuție rezultată din combinarea a *K* distribuții normale (*K* ales de utilizator, pe baza experienței sau prin încercări), nu se cunosc valorile $P(y=1)=\pi_1,...,P(y=K)=\pi_K$, nu se cunosc parametrii acestor distribuții, adică toate necunoscute/parametrii pot fi grupate formal în $\boldsymbol{\theta}=[\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_K,\boldsymbol{\Sigma}_1,...,\boldsymbol{\Sigma}_K,\pi_1,...,\pi_K]^T$. Fie o inițializare $\boldsymbol{\theta}^0$. Atunci, la fiecare iterație *k* a algoritmului EM sunt efectuați următorii pași:

Pasul 1. (**Expectation**) Se calculează pentru fiecare instanță cel mai probabil grup de care aparține prin calculul indicatorilor:

$$R_{ij}^{k-1}(\boldsymbol{\theta}^{k-1})=P(y_i=j\mid \mathbf{X}=\mathbf{x}_i,\boldsymbol{\theta}^{k-1})=\frac{fdp(\mathbf{x}_i,\boldsymbol{\mu}_j^{k-1},\boldsymbol{\Sigma}_j^{k-1})\pi_j^{k-1}}{\sum_{j=1}^{K}fdp(\mathbf{x}_i,\boldsymbol{\mu}_j^{k-1},\boldsymbol{\Sigma}_j^{k-1})\pi_j^{k-1}}, \text{ pentru } i=1,n, j=1,K. \quad (10)$$

Pasul 2. (**Maximization**) Se actualizează parametrii $\boldsymbol{\theta}^k$ folosind estimatorul MLE:

$$\begin{aligned}
&\boldsymbol{\mu}_j^k=\sum_{i=1}^{n}\omega_i^j\mathbf{x}_i, \omega_i^j=\frac{R_{ij}^{k-1}}{\sum_{i=1}^{n}R_{ij}^{k-1}},\\
&\boldsymbol{\Sigma}_j^k=\sum_{i=1}^{n}\omega_i^j(\mathbf{x}_i-\boldsymbol{\mu}_j^k)^T(\mathbf{x}_i-\boldsymbol{\mu}_j^k), \text{ pentru } i=1,n, j=1,K \quad (11)\\
&\pi_j^k=\frac{1}{n}\sum_{i=1}^{n}R_{ij}^{k-1}.
\end{aligned}$$

Algoritmul EM poate fi considerat terminat atunci când modificările în parametrii estimați ai modelului GMM dintre două iterații succesive sunt suficient de mici. După finalizarea algoritmului EM, relația (10) poate fi utilizată pentru a grupa instanțele de date. Astfel, fiecare instanță va "activa" toate f.d.p. normale care compun modelul GMM cu probabilitățile calculate cu relația (2). Apoi, pe baza probabilităților π_j, poate fi utilizată relația (10) pentru a calcula, pentru fiecare instanță valorile $R_{ij}(\boldsymbol{\theta})=P(y_i=j\mid \mathbf{X}=\mathbf{x}_i,\boldsymbol{\theta})$. Indicele *j* care produce cel mai mare coeficient $R_{ij}(\boldsymbol{\theta})$ va indica grupul (sau distribuția normală/f.d.p) de care instanța este cel mai probabil să aparțină.

Ca exemplificare a pașilor algoritmului EM prezentați anterior, fie mulțimea instanțelor din plan ($d=2$) având coordonatele $D=\{\mathbf{x}_1=[-1\ -1]^T,\mathbf{x}_2=[2\ 1]^T,\mathbf{x}_3=[1\ 2]^T\}, n=3$, pentru care se dorește gruparea prin algoritmul EM aplicat unei distribuții GMM bimodale ($K=2$) caracterizată de distribuții normale bivariate. Practic, vor rezulta două grupuri la care instanțele aparțin cu o probabilitate mai mare sau mai mică. Inițializarea algoritmului este $\boldsymbol{\mu}_1^0=[-1\ -1]^T,\boldsymbol{\mu}_2^0=[1\ 1]^T,\boldsymbol{\Sigma}_1^0=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\boldsymbol{\Sigma}_2^0=\begin{bmatrix}0{,}5 & 0\\ 0 & 0{,}5\end{bmatrix},\pi_1^0=0{,}5,\pi_2^0=0{,}5$.

La prima iterație, Pasul 1 calculează în primă fază, cu ajutorul relației (2), pentru toate instanțele, funcțiile de densitate de probabilitate. Astfel $fdp(\mathbf{x}_1,\boldsymbol{\mu}_1^0,\boldsymbol{\Sigma}_1^0)=0{,}0796$, $fdp(\mathbf{x}_1,\boldsymbol{\mu}_2^0,\boldsymbol{\Sigma}_2^0)=0{,}0001$, $fdp(\mathbf{x}_2,\boldsymbol{\mu}_1^0,\boldsymbol{\Sigma}_1^0)=0{,}0001$, $fdp(\mathbf{x}_2,\boldsymbol{\mu}_2^0,\boldsymbol{\Sigma}_2^0)=0{,}1171$, $fdp(\mathbf{x}_3,\boldsymbol{\mu}_1^0,\boldsymbol{\Sigma}_1^0)=0{,}0001, fdp(\mathbf{x}_3,\boldsymbol{\mu}_2^0,\boldsymbol{\Sigma}_2^0)=0{,}1171$. Din

care rezultă conform relației (10): $R_{11}^0 = 0{,}9987, R_{12}^0 = 0{,}0013, R_{21}^0 = 0{,}0010, R_{22}^0 = 0{,}9990$, $R_{31}^0 = 0{,}0010, R_{32}^0 = 0{,}9990$.

Pasul doi al primei iterații produce conform (11), pe rând, $\omega_1^1 = 0{,}9980, \omega_1^2 = 0{,}0007, \omega_2^1 = 0{,}0010, \omega_2^2 = 0{,}4997, \omega_3^1 = 0{,}0010, \omega_3^2 = 0{,}4997$. Apoi, $\boldsymbol{\mu}_1^1 = [-0{,}9949 \; -0{,}9949]^T, \boldsymbol{\mu}_2^1 = [1{,}4983 \; 1.4983]^T$. Mai rezultă $\pi_1^1 = 0{,}3336, \pi_2^1 = 0{,}6664$ și $\boldsymbol{\Sigma}_1^1 = \begin{bmatrix} 0{,}0265 & 0{,}0265 \\ 0{,}0265 & 0{,}0265 \end{bmatrix}, \boldsymbol{\Sigma}_2^1 = \begin{bmatrix} 0{,}5080 & 0{,}5080 \\ 0{,}5080 & 0{,}5080 \end{bmatrix}$.

Cei doi pași descriși mai sus se reiau la iterația următoare și pot fi continuați până la modificări nesemnificative în parametrii $\boldsymbol{\theta}^k$ ai modelului GMM .

Programul Matlab de mai jos exemplifică soluția unei probleme de grupare pe un set de date bidimensional compus din două distribuții normale al căror grad de suprapunere (de combinare) poate fi variat prin modificarea centrelor și dispersiilor acestor distribuții. Rezultă deci o distribuție GMM bimodală bi-variată. Funcția Matlab *fitgmdist*() conduce la calculul parametrilor unui număr de grupuri/clase folosind algoritmul EM. Rezultatele sunt vizualizate în Fig. 3.

```
COD 1
% ------------------exempluEM.m---------------------- %
MU1 = [1 0.5];
SIGMA1 = [4 0; 0 .5];
MU2 = [-2 -2]; % MU2=[-3 -5]
SIGMA2 = [1 0; 0 1];
dataSet = [mvnrnd(MU1,SIGMA1,1000);mvnrnd(MU2,SIGMA2,1000)];

scatter(dataSet(:,1),dataSet(:,2),10,'.')
hold on

modelGMM=fitgmdist(dataSet,2); % model GMM cu 2 componente in 2 dimensiuni
modelGMM.mu, % afisare medii
modelGMM.Sigma, % afisare matrice de covariante
h = ezcontour(@(x,y)pdf(modelGMM,[x y]),[-8 6],[-8 6]);

%evalueaza o instanta (x1 x2) in cele doua pdf-uri aferente celor 2 componente
pct=[-3 -5]';d=2;
modelGMM.pdf([-3 -5])
prob1=1/((2*pi)^(d/2)*sqrt(det(modelGMM.Sigma(:,:,1)))) * exp( -0.5*(pct-
modelGMM.mu(1,:)')'*inv((modelGMM.Sigma(:,:,1)))*(pct-modelGMM.mu(1,:)') )
prob2=1/((2*pi)^(d/2)*sqrt(det(modelGMM.Sigma(:,:,2)))) * exp( -0.5*(pct-
modelGMM.mu(2,:)')'*inv((modelGMM.Sigma(:,:,2)))*(pct-modelGMM.mu(2,:)') )

% grupare pe fisheriris dataset
clear
load fisheriris
dataSet=meas(:,1:2);
scatter(dataSet(:,1),dataSet(:,2),15,'o')
hold on
modelGMM= fitgmdist (dataSet,3); % model GMM cu 3 componente (distributii) in 2 dimensiuni
modelGMM.mu, % afisare medii
modelGMM.Sigma, % afisare matrice de covariante
h = ezcontour(@(x,y)pdf(modelGMM,[x y]),[4 8],[2 5]);
%-------------------------------------------------------------------------------%
```

Liniile 17–23 sugerează o posibilă evaluare a probabilităților de apartenență a unei instanțe, la cele două grupuri rezultate în urma grupării cu EM. Evaluarea utilizează două modele de distribuție normală bivariată de forma (2), cu parametrii medie și matrice de covarianță ai fiecărei distribuții (f.d.p.) preluați din câmpurile obiectului *modelGMM*.

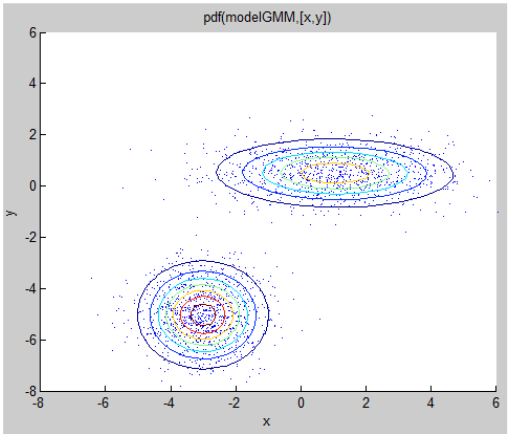

Fig. 3 Relativă la soluția de grupare cu algoritmul EM folosind modele GMM.

Partea a doua a codului (începând la linia 26) aplică algoritmul EM la gruparea unor date care reprezintă măsurători ale unor parametri care corespund la trei specii de flori de iris: lungimea și lățimea sepalelor și petalelor. Deși setul de date poate fi folosit în probleme de clasificare datorită existenței etichetelor de apartenență la o anumită clasă, pentru problema de grupare va fi ignorat atributul de clasă și se va încerca doar rezolvarea problemei de grupare. Pentru o mai bună vizualizare, gruparea cu EM folosind model GMM va fi efectuată doar pe primele două atribute (măsurători). Rămâne ca și exercițiu pentru cititor afișarea acestor rezultate.

Ca și remarci finale la algoritmul EM, pot fi menționate:

- Cu fiecare rulare nouă sunt obținute rezultate diferite, datorită inițializărilor aleatoare.
- Numărul de grupuri (clustere) trebuie ales în baza unor analize prealabile asupra datelor și reprezintă un parametru de inițializare. În ambele exemple din cod, a fost cunoscut numărul de grupuri care a generat distribuțiile supuse modelării cu GMM. În realitate, datele nu sunt etichetate și selecția numărului de grupuri implică proceduri euristice de vizualizare/prelucrare a datelor, în special atunci când datele sunt mai în mai multe dimensiuni decât cele 3 spațiale ușor vizualizabile. Pot fi folosite de exemplu tehnici de reducere (compresie) a dimensionalității datelor, cum ar fi analiza componentelor principale, tehnici de proiecție, sau rețele neuronale de tip autoencoder.
- O variantă de k-means++ poate fi utilizată și la inițializarea parametrilor algoritmului EM (Matlab 2015).
- Coeficienții π_j ai obiectului GMM pot fi accesați din vectorul câmpului *ComponentProportion*.

Exemplu 1. Programul Matlab redat mai jos exemplifică gruparea unor date provenind din măsurători ale lungimii și lățimii sepalelor provenind de la trei specii de flori de iris, folosind *k-means* și EM cu model GMM. Rezultatele comparative sunt afișate în Fig. 4.

```
1  COD 2
2  % ------------------exemplu_kmeans_EM.m---------------------- %
3  clear
```

```
load fisheriris
dataSet=meas(:,1:2);

scatter(dataSet(:,1),dataSet(:,2),15,'o'),
xlabel('lungime petale'),ylabel('latime petale'),title('datele negrupate')
hold on
nrCentre=input('Numar centre: ')
[initX,initY]=ginput(nrCentre);

[idxs,centreKM]=kmeans(dataSet,nrCentre,'Start',[initX,initY])

x = min(dataSet(:,1)):0.01:max(dataSet(:,1));
y = min(dataSet(:,2)):0.01:max(dataSet(:,2));
[xx,yy]=meshgrid(x,y);
gridDataSet=[xx(:),yy(:)];
% o singura iteratie a kmeans face atribuirea tuturor punctelor din reteaua
% de puncte, celor doua grupuri cu centrele deja calculate la rularea
% anterioara kmeans. poate fi interpretata ca o clasificare
idxGrid = kmeans(gridDataSet,nrCentre,'MaxIter',1,'Start',centreKM);

gscatter(gridDataSet(:,1),gridDataSet(:,2),idxGrid,[0,0.8,0.8;0.8,0,0.8;0.8,0.8,0]),
hold on,gscatter(dataSet(:,1),dataSet(:,2),idxs),hold on,plot(centreKM(:,1),centreKM(:,2),'k*'),
xlabel('lungime sepale'),ylabel('latime sepale'),title('k-means')

% este clar din figura ca desi apar doua grupuri clare de date, kmeans cu
% doua centre nu produce o grupare corecta

%% cu GMM
modelGMM= fitgmdist (dataSet,2); % model GMM cu 2 componente in 2 dimensiuni
modelGMM.mu, % afisare medii
modelGMM.Sigma, % afisare matrice de covariante
figure
idxGridGmm=cluster(modelGMM,gridDataSet);
idxGmmData=cluster(modelGMM,dataSet);
gscatter(gridDataSet(:,1),gridDataSet(:,2),idxGridGmm,[0,0.8,0.8;0.8,0,0.8;0.8,0.8,0]),
hold on,gscatter(dataSet(:,1),dataSet(:,2),idxGmmData),hold on
h = ezcontour(@(x,y)pdf(modelGMM,[x y]),[4 8],[2 4.5]);hold on,
xlabel('lungime sepale'),ylabel('latime sepale'),title('EM cu GMM')
% afisare coeficienti de silueta
figure,silhouette(dataSet,idxs)
figure,silhouette(dataSet,idxGmmData)
%-------------------------------------------------------------------------------------%
```

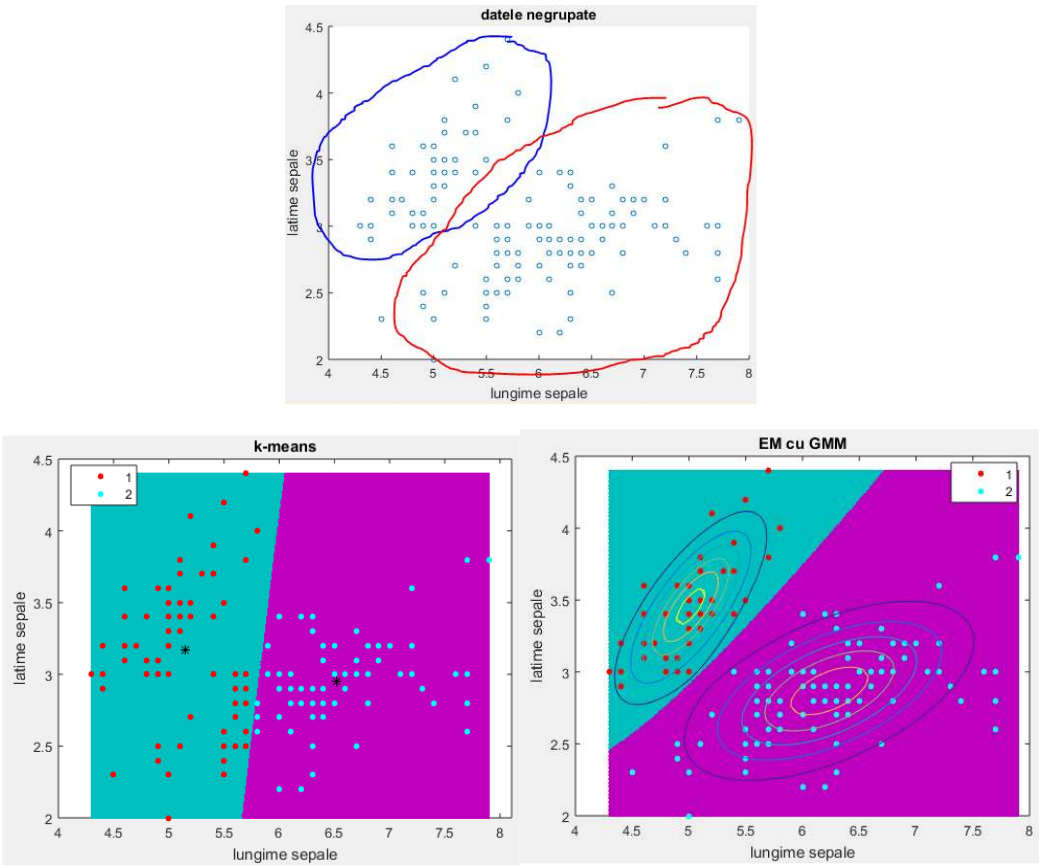

Fig. 4 Soluția de grupare folosind k-means (stânga jos) și EM cu GMM (dreapta jos). În partea de sus pot fi vizualizate datele originale negrupate.

Poate fi observat în partea de sus a Fig. 4 anterioare că în datele negrupate pot fi evidențiate două grupuri, în interiorul zonelor conturate cu albastru respectiv roșu. Pentru gruparea folosind atât k-means cât și EM, sunt de dorit deci două grupuri, din analiza inițială a datelor.

Soluția de grupare cu *k-means* iterată pornind de la două centre inițiale citite de pe grafic folosind mausul conduce la graficul din Fig. 4 stânga jos în care poate fi observat că gruparea s-a efectuat în mod nenatural. Pe când soluția de grupare cu EM furnizată și afișată în Fig. 4 dreapta jos, este mai naturală.

În ambele soluții de grupare, este utilizată o rețea foarte fină de puncte pentru a ilustra grafic granița care delimitează apartanența la unul din cele două clustere: prin distanța euclideană față de cele două centre folosită la grupare în cazul *k-means*, respectiv prin probabilitatea de apartenență la una din cele două distribuții în cazul EM. Este interesant de remarcat faptul că centrele clusterelor (centrele k-means respectiv vârfurile funcțiilor de densitate de probabilitate ale distribuțiilor EM) în cazul celor două abordări nu diferă prea mult. Totuși, granițele care delimitează grupurile identificate au orientări vizibil diferite, cea rezultată în urma grupării folosind EM având evident caracterul mai natural.

Afișarea coeficienților de siluetă pentru instanțele grupate cu EM și respectiv cu k-means în Fig. 5 indică rezultate ușor de interpretat pentru acest exemplu particular. Cu precizarea că aceste grafice vor diferi cu fiecare rerulare a procesului de grupare cu k-means și respectiv cu EM. Valoarea coeficienților este mai mică pentru instanțele din primul grup (inclusiv valori negative) în cazul grupării cu k-means (Fig. 5) decât în cazul grupării folosind EM. În plus, în cazul primului grup obținut cu EM, valorile coeficienților tuturor instanțelor sunt relativ mari. Analiza indică faptul că soluția EM este mai buna. Pe de altă parte, pentru instanțele din cel de-al doilea grup, valorile coeficienților sunt mai mari în cazul soluției k-means decât în cazul soluției EM, pentru care apar inclusiv valori negative ale coeficienților de siluetă. Rezultatul

indică faptul că sunt prea puține sau prea multe grupuri pentru instanțele arondate grupului 2. Fig. 5 dreapta aferentă grupării EM indică instanțe mai izolate în raport cu centrul determinat al grupului, motiv pentru care probabil ar fi fost necesare grupuri (centre) suplimentare.

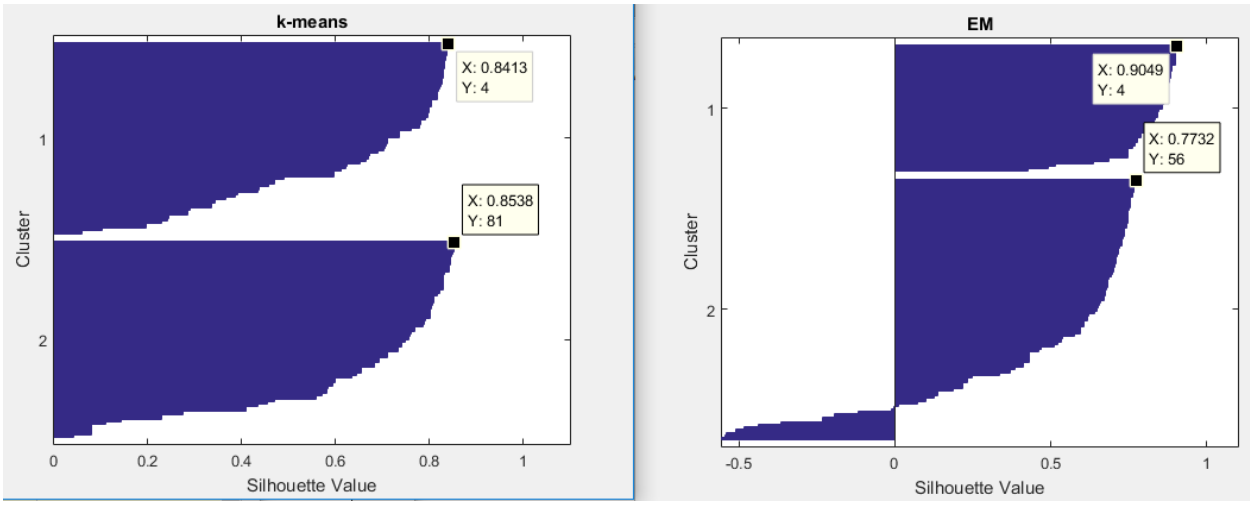

Fig. 5 Coeficienții de siluetă ai instanțelor la grupare cu k-means (stânga) și respectiv cu EM (dreapta).

Exerciții propuse:

1) Realizați o documentație a funcțiilor *kmeans* și *fitgmdist* din Matlab care să cuprindă o descriere a funcționalităților, prezentarea argumentelor de intrare și de ieșire, informații despre algoritmul care stă la baza implementării, precum și diverse exemple de rulare și afișare a datelor, pe alte seturi de date, chiar de dimensiuni mai mari.
2) Scrieți o funcție Matlab similară cu *kmeans* care să conțină o implementare proprie a algoritmului k-means cu inițializarea centrelor folosind algoritmul k-means++.
3) Setul Wisconsin Breast Cancer Dataset (WBCD) (Lichman 2013) de pe UCI Machine learning Repository este un set de date folosit în probleme de grupare și clasificare. Setul conține instanțe cu 32 de atribute de intrare calculate din imagini digitale ale unor probe de biopsie recoltate care descriu caracteristicile nucleelor celulare prezente în imagini. Setul de date este pregătit în format .dat care poate fi importat în Matlab. Toate aceste atribute sunt numere reale pe diverse scări de amplitudine. În total există 569 de instanțe. Atributul de clasificare de ieșire este 'M' – malign sau 'B' – benign și stabilește dacă proba recoltată este canceroasă sau nu. Acest atribut de clasă poate fi convertit la valoare numerică (1, respectiv 0, sau 1 respectiv -1), sau orice alte valori. În cazul problemei de grupare nu interesează clasificarea instanțelor existente deci etichetele vor fi ignorate.
   Prima coloană din setul de date reprezintă un număr de identificare (ID) al instanței, a doua coloană reprezintă atributul de ieșire la clasificare pentru învățare supervizată (malign sau benign), celelalte atribute de învățare încep cu coloana a treia.
   Sunt extrase doar primele două atribute de învățare (coloanele 3 și 4) și sunt afișate corespunzător în Fig. 4 de mai jos. Aceste atribute reprezintă raza medie a nucleelor celulare și respectiv textura acestora măsurată ca deviația standard a valorii nuanțelor de gri ale pixelilor.
   În Matlab există un set de date similar numit *cancer_dataset* care are un număr mai mare de instanțe (699), dar mai puține atribute care sunt în plus și normalizate. Și acest set de date poate fi încărcat și mai apoi utilizat pentru probleme de grupare și clasificare a datelor.

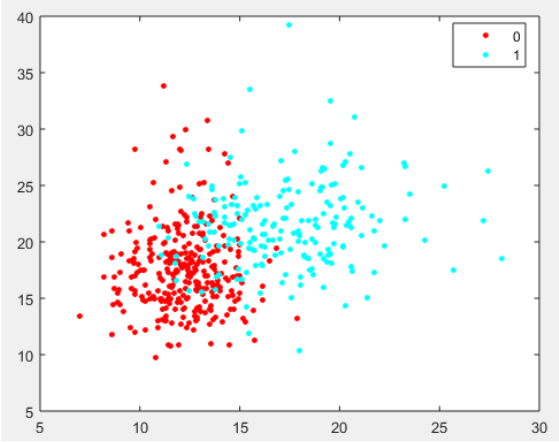

Fig. 4 Instanțele aferente primelor două atribute din setul de date WBCD clasificate corespunzator ca benign (0) si malign (1).

Se cere să se testeze algoritmul EM folosind atât funcția de grupare *fitgmdist*() din Matlab cât și în variantă proprie de implementare în Matlab, pentru gruparea datelor folosind un model GMM. Se vor observa și documenta rezultatele.

Bibliografie

Hastie T., R. Tibshirani, J. Friedman (2009). The elements of statistical learning: data mining, inference, and prediction, 2nd edition, Springer-Verlag.

Haykin S. (2009). Neural networks and learning machines, 3rd Edition, Prentice Hall, Upper Saddle River, NJ.

Arthur D., S. Vassilivitskii (2007). K-means++: the advantages of careful seeding. In Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, USA, pp. 1027–1035.

Poczos B., A. Singh (2014). Introduction to machine learning. Clustering and EM (lectures on). Carnegie Mellon School of Computer Science.

Lichman M. (2013). UCI machine learning repository [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29]. Irvine, CA: University of California, School of Information and Computer Science.

Matlab (2015). Statistics and machine learning toolbox. Mathworks Inc., Natick, MA.

Andrews N. O., E. A. Fox (2007). Recent developments in document clustering. Technical report TR-07-35. Deparment of Computer Science, Virginia Tech.

Baldi P., G. Hatfield (2002). DNA microarrays and gene expression, Cambridge University Press, MA.

Dhillon I. S., S. Mallela, R. Kumar (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, vol. 3, pp 1265–1287.

Lukashin A. V. , M. E. Lukashev, R. Fuchs (2003). Topology of gene expression networks as revealed by data mining and modeling. *Bioinformatics*, vol. 19 , no. 15, pp. 1909–1916.

Philibin J., O. Chum, M. Isard, J. Sivic, A. Zisserman (2007). Object retrieval with large vocabularies and fast spatial matching. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, pp. 1–8.

Shi J., J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905.

M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, C. Potter (2003). Discovery of climate indices using clustering. In Proc. Of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington D.C., pp. 446–455.

Slatkin M., L. Excoffier (1996). Testing for linkage disequilibrium in genotypic data using the Expectation-Maximization algorithm. *Heredity*, vol 76, pp. 377–383.

Fallin D., N. J. Schork (2000). Accuracy of haplotype frequency estimation for Biallelic Loci, via the Expectation-Maximization algorithm for unphased diploid genotype data. *The American Journal of Human Genetics*, vol. 67, no. 4, pp. 947–959.

Dayan P., G. E. Hinton (2006). Using Expectation-Maximization for reinforcement learning. *Neural Computation*, vol. 9, no. 2, pp 271–278.

Ramme A. J., N. DeVries, N. A. Kallemyn, V. A. Magnotta, N. M. Grosland (2009). Semi-automated phalanx bone segmentation using the Expectation Maximization algorithm. *Journal of Digital Imaging*, vol. 22, no. 5, pp. 483–491.

Frenkel L., M. Feder (1999). Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking. *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 306–320.

Zhang Y., M. Brady, S. Smith (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57.

Fatakdawala H., J. Xu, A. Basavanhally, G. Bhanot, S. Ganesan, M. Feldman, J. E. Tomaszewski, A Madabhushi (2010). Expectation–Maximization-driven geodesic active contour with overlap resolution (EMaGACOR): application to lymphocyte segmentation on breast cancer histopathology. IEEE Transcations on Biomedical Engineering, vol. 57, no. 7, pp. 1676–1689.

Zafrir N., A. Solodky, A. Ben-Shlomo, I. Mats, R. Nevzorov, A. Battler, A Gutstein (2012). Feasibility of myocardial perfusion imaging with half the radiation dose using ordered-subset expectation maximization with resolution recovery software. *Journal of Nuclear Cardiology*, vol. 19, no. 4, pp. 704–712.

Marinakis D., G. Dudek, D. J. Fleet (2005). Learning sensor network topology through monte carlo expectation maximization. In Proc. of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 4581–4587.