Inferență statistică în ML

Cap 11. K-means. Expectation maximization.

Modele de combinare a gaussienelor.

May 11, 2025

K-means clustering

- problema constă în identificarea grupurilor (clusterelor) formate de puncte într-un spațiu multidimensional
- considerăm un set de date $\{x_1, x_2, \ldots x_N\}$ constînd în N observații într-un spațiu euclidian D dimensional
- vrem să partiționăm datele în K clustere (K este dat)
- un cluster cuprinde punctele pentru care distanțele între punctele din cluster sunt mai mici decât distanțele spre punctele din afara clusterului
- asociem cu fiecare cluster un vector D-dimensional μ_k, $k = 1 \ldots K$, denumit **vector prototip**; acesta va reprezenta centrul clusterului
- scopul este găsirea unei asocieri între fiecare vector și un cluster, precum și un set de vectori μ_k, astfel ca distanța dintre fiecare punct și vectorul μ_k asociat clusterului său să fie minimă

K-means clustering: notații și funcția de optimizat

- pentru fiecare punct x_n introducem un set de variabile indicator, $r_{nk} \in \{0, 1\}$, unde $k = 1 \dots K$ descrie cărui cluster îi este asociat punctul x_n
- dacă punctul x_n este asociat clusterului k atunci $r_{nk} = 1$ și $r_{nj} = 0$ pentru toți $j = 1 \dots K$ unde $j \neq k$
- funcția de optimizat (**distortion measure**):

$$\min J(x, r, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2 \tag{1}$$

- scopul este găsirea lui $\{r_{nk}\}$ și $\{\mu_k\}$ astfel ca $J(\cdot)$ să fie minimă

K-means clustering: proces iterativ

- putem optimiza individual în doi pași separați
  - optimizarea lui J doar pentru găsirea r_{nk} optime, cu μ_k fixat (Expectation step)
  - optimizarea lui J separat pentru găsirea μ_k, cu r_{nk} fixat (Maximization step)
- repetăm cei doi pași până la convergență (centroizii μ_k nu se mai schimbă)
- optimizarea pentru r_{nk} dacă μ_k e fixat se face prin găsirea celei mai mici distanțe $\|x_n - \mu_k\|^2$ pentru fiecare n - asignarea unui punct celui mai apropiat centru de cluster

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_n - \mu_j\|^2 \\ 0 & \text{în caz contrar} \end{cases} \tag{2}$$

K-means clustering: maximizarea

- pentru cazul în care fixăm r_{nk}, pentru a determina optimul μ_k pentru $J(\cdot)$, vom egala prima derivată cu zero:

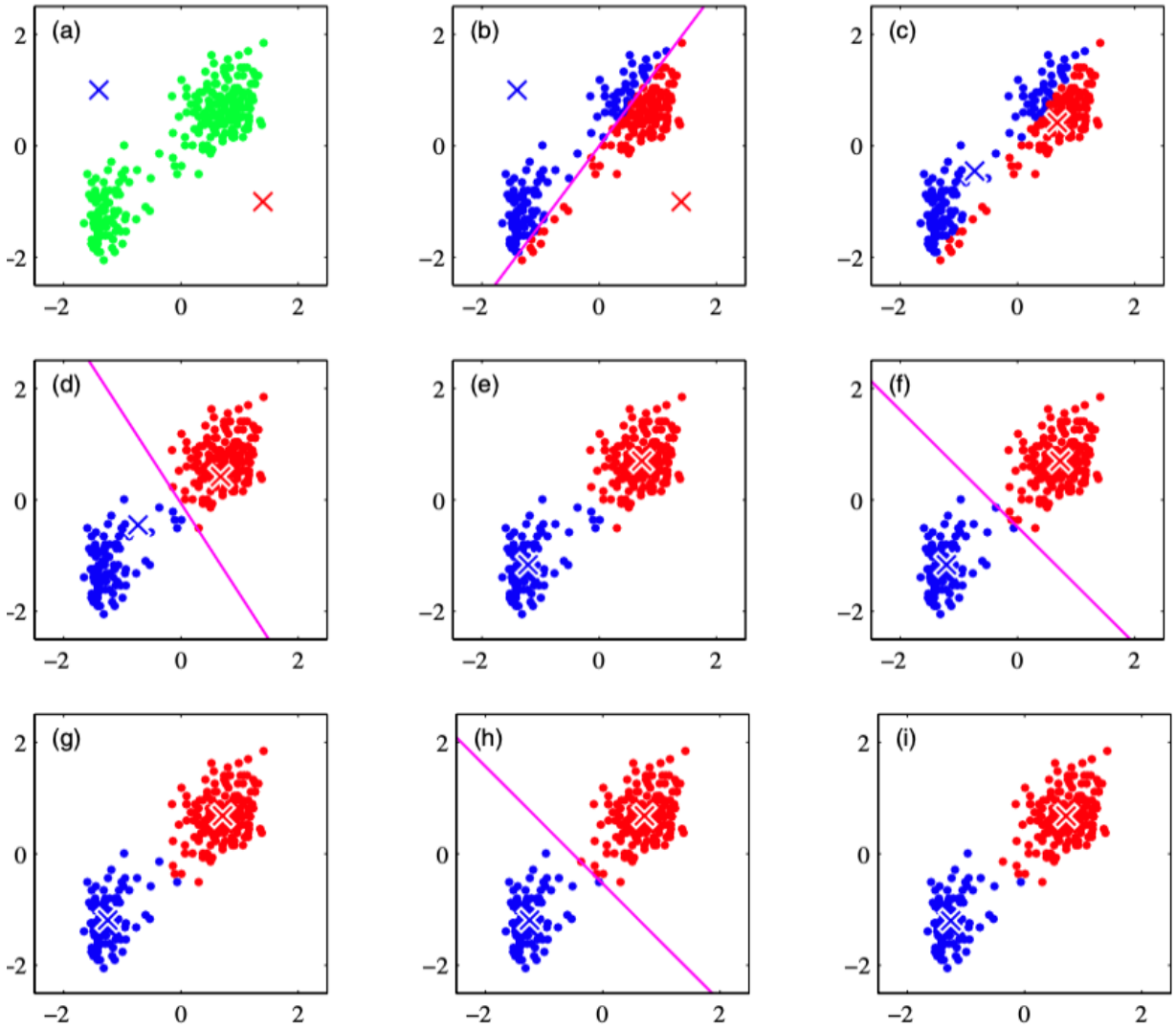
$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0 \tag{3}$$

- de unde avem expresia pentru μ_k

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}} \tag{4}$$

- numitorul este numărul punctelor asociate clusterului k, practic se face media punctelor clusterului k

Old Faithful dataset[1] convergență în 8 pași



[1] Bishop, Christopher M., "Pattern Recognition and Machine Learning", New York, Springer, 2006: figura 9.1

Avantaje și dezavantaje

- algoritmul inițializează aleator μ_k
- pentru convergență mai rapidă, o mai bună inițializare se face prin inițializarea nu cu valori aleatoare a centroizilor μ_k, ci prin inițializarea lor aleatorie cu puncte deja existente în dataset
- K-means se folosește pentru inițializarea algoritmului EM al modelului de gaussiene combinate (Gaussian mixture model)
- algoritmul presupune calculul, la fiecare pas, a distanțelor de la centroizi la fiecare punct - timp O(NK); îmbunătățiri:
  - precalcularea unei structuri de date precum un arbore, astfel ca toate punctele alăturate să se găsească deja într-un subarbore
  - folosirea inegalității triunghiului pentru estimarea distanțelor, evitând recalcularea unor distanțe
- pentru K-means, fiecare punct este asociat cu un singur centroid (abordare "hard")
- algoritmul EM prespune ca fiecare punct să fie asociat cu toți centroizii, însă în grade diferite, ce reflectă nivelul de incertitudine (abordare probabilistă, "soft")

Exemplu cu aruncări a două monezi[2]

- avem două monezi A și B, cu probabilitățile de a obține head θ_A și θ_B necunoscute
- ne interesează să aflăm probabilitățile θ_A și θ_B
- avem rezultatul următorului experiment (50 de aruncări în total):

Moneda	Aruncări	# heads pt. A	# heads pt. B
B	HT TT HH TH TH	0	5
A	HH HH TH HH HH	9	0
A	HT HH HH HT HH	8	0
B	HT HT TT HH TT	0	4
A	TH HH TH HH TH	7	0

[2] https://www.baeldung.com/cs/expectation-maximization-technique

Calculul direct pentru θ_A și θ_B

- calculăm proporția de heads pentru fiecare monedă în parte

$$\theta_A = \frac{\#\text{ heads observat pentru moneda A}}{\#\text{ total de aruncări pentru A}} = \frac{24}{30} = 0.8 \quad (5)$$

$$\theta_B = \frac{\#\text{ heads observat pentru moneda B}}{\#\text{ total de aruncări pentru B}} = \frac{9}{20} = 0.45 \quad (6)$$

- ce facem dacă pierdem identitatea monezilor (etichetarea lor)?
- pe lângă parameterul θ vom mai avea o variabilă latentă t - identitatea monezii

Exemplu: observațiile reale

- nu vom cunoaște identitatea monezii folosite la aruncarea E

Aruncarea	Secvența E de aruncări	# heads
1	HT TT HH TH TH	5
2	HH HH TH HH HH	9
3	HT HH HH HT HH	8
4	HT HT TT HH TT	4
5	TH HH TH HH TH	7

Expectation step

- pornind de la datele extragerilor, putem, pentru fiecare din cele 5 extrageri, să presupunem care dintre cele două monezi a fost folosită - pentru care monedă rezultatul observat E este cel mai plauzibil
- nu vom face o asociere hard (K-means) ci una soft, asociem o probabilitate de asociere a fiecărei monede cu secvența observată, $P(E|t_A)$ respectiv $P(E|t_B)$
- putem calcula aceste probabilități dacă presupunem că fiecare aruncare E este realizarea unei distribuții binomiale
- probabilitatea să iasă x head-uri succesive din x aruncări:

$$P(E|t_A) = P(HTTTHHTHTH|t_A) = C_n^x \theta_A^x (1-\theta_A)^{1-x} \quad (7)$$

$$P(E|t_B) = P(HTTTHHTHTH|t_B) = C_n^x \theta_B^x (1-\theta_B)^{1-x} \quad (8)$$

- avem nevoie de θ_A și θ_B, pentru primul pas le inițializăm aleator, $\theta_A = 0.6$ și $\theta_B = 0.5$

Expectation step: regula lui Bayes

- ne interesează să calculăm probabilitățile de asociere cu fiecare monedă A sau B dacă observăm secvența E, adică $P(t_A|E)$ respectiv $P(t_B|E)$
- putem calcula asta din regula lui Bayes:

$$P(t_A|E) = \frac{P(E|t_A)P(t_A)}{P(E)} = \frac{P(E|t_A)P(t_A)}{P(E|t_A)P(t_A) + P(E|t_B)P(t_B)} \quad (9)$$

$$P(t_B|E) = \frac{P(E|t_B)P(t_B)}{P(E)} = \frac{P(E|t_B)P(t_B)}{P(E|t_A)P(t_A) + P(E|t_B)P(t_B)} \quad (10)$$

- inițial, presupunem că cele două monede sunt echiprobabile, $P(t_A) = P(t_B) = 0.5$ (prior probabilities)
- cele două evenimente sunt complementare, fie aruncăm cu moneda A fie cu moneda B, deci întotdeauna $P(t_A) + P(t_B) = 1$

Expectation step: calculul realizărilor așteptate

- pentru fiecare monedă, având probabilitatea asociată cu secvența de aruncări, $P(t_A|E)$, putem calcula valoarea așteptată a numărului de heads și tails:

$$\text{heads}_A = \textit{textnr.headsdinE} \cdot P(t_A|E) \tag{11}$$

$$\text{tails}_A = (10 - \text{nr. heads din E}) \cdot P(t_A|E) \tag{12}$$

$$\text{heads}_B = \text{nr. heads din E} \cdot P(t_B|E) \tag{13}$$

$$\text{tails}_B = (10 - \text{nr. heads din E}) \cdot P(t_B|E) \tag{14}$$

- putem pune rezultatele sub formă tabelară:

Realizările așteptate la primul pas

	# Heads	P(t_A | E)	P(t_B | E)	# heads A	# tails A	# heads B	# tails B
0	5	0.449149	0.550851	2.25	2.25	2.75	2.75
1	9	0.804986	0.195014	7.24	0.80	1.76	0.20
2	8	0.733467	0.266533	5.87	1.47	2.13	0.53
3	4	0.352156	0.647844	1.41	2.11	2.59	3.89
4	7	0.647215	0.352785	4.53	1.94	2.47	1.06
5			Total	21.30	8.57	11.70	8.43

- observați că $P(t_A)$ și $P(t_B)$ se schimbă:

$$P(t_A) = \frac{21.30 + 8.57}{21.30 + 8.57 + 11.70 + 8.43} = \frac{29.87}{50.0} = 0.597 \quad (15)$$

$$P(t_B) = \frac{11.70 + 8.43}{21.30 + 8.57 + 11.70 + 8.43} = \frac{20.13}{50.0} = 0.403 \quad (16)$$

- devin foarte aproape de ceea ce știam deja înainte să pierdem etichetările!

Maximization step

- folosind noile date pentru numărul așteptat de heads, putem calcula noi valori estimate pentru parametrii θ_A și θ_B (probabilitățile să iasă head pentru cele două monezi):

$$\theta_A = \frac{21.30}{21.30 + 8.57} = 0.713 \tag{17}$$

$$\theta_B = \frac{11.70}{11.70 + 8.43} = 0.581 \tag{18}$$

- odată re-estimate valorile parametrilor, putem să reluăm cei doi pași în mod repetat până când acestea nu se mai modifică

Mixture model

- ceea ce facem de fapt este să potrivim niște distribuții cu parametri necunoscuți peste date
- datele observate sunt generate de un mix de distribuții
- funcția de probabilitate a acelui mix de distribuții (ce încercăm să maximizăm) este:

$$P(X) = \sum_{k=1}^{K} \pi_k N(X|\theta_k) \tag{19}$$

- avem un mix de K distribuții (superpoziție de gaussiene), fiecare cu setul său de parametri θ_k
- variabila latentă t poate lua valori $t_k \in \{0, 1\}$ și selectează care distribuție produce rezultatul X, deci $\sum_k t_k = 1$; sunt K stări posibile pentru t în funcție de care poziție e diferită de zero

Prior, likelihood și posterior

- distribuția compusă $P(X, t)$ definește probabilitatea de a observa datele X în condițiile existenței variabilei latente t
- avem câteva definiții pe care le putem ilustra cel mai bine folosind regula lui Bayes
- observăm datele X generate de unul din modele, modelele funcționând cu parametrii θ
- probabilitatea evenumentului simultan (date, model) este:

$$P(X, t) = P(X|t) \cdot P(t) = P(t|X) \cdot P(X) \tag{20}$$

- $P(X|t)$ se numește **likelihood**, probabilitatea de a observa datele dat fiind un anumit model
- $P(t)$ se numește **probabilitatea apriori**, sau prior, este probabilitatea ca intern datele să fie generate de modelul t
- $P(t|X)$ se numește **probabilitatea aposteriori**, probabilitatea să fi fost ales modelul t dacă s-au observat datele X

Variabila latentă t

- distribuția marginală a lui t este specificată de coeficienții de mix π_k:

$$p(t_k = 1) = \pi_k \tag{21}$$

- unde parametrii π_k verifică $0 \leq \pi_k \leq 1$ și $\sum_k \pi_k = 1$
- distribuția lui t poate fi scrisă astfel (un singur t_k nenul):

$$p(t) = \prod_{k=1}^{K} \pi_k^{t_k} \tag{22}$$

- probabilitatea condiționată a lui X pentru o valoare particulară a lui t este o gaussiană:

$$P(x|t_k = 1) = \mathcal{N}(X|\theta_k) \tag{23}$$

- probabilitatea marginală a lui X poate fi descompusă folosind suma peste toate valorile posibile pentru t:

$$P(X) = \sum_t P(t)P(X|t) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\theta_k) \tag{24}$$

Calculul pentru Expectation

- dacă avem mai multe observații $x_1 \dots x_N$, deoarece am reprezentat distribuția marginală în forma $P(X) = \sum_t P(X, t) = \sum_t P(t)P(X|t)$, atunci vom avea pentru fiecare valoare observată x_n o variabilă t_n
- ne interesează probabilitatea aposteriori, ca $t_k = 1$ când am observat datele X:

$$\gamma(t_k) = P(t_k = 1|X) = \frac{P(t_k = 1)P(X|t_k = 1)}{\sum_{j=1}^{K} P(t_j = 1)P(X|t_j = 1)} \quad (25)$$

$$= \frac{\pi_k \mathcal{N}(X|\theta_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(X|\theta_j)} \quad (26)$$

- s-a aplicat regula lui Bayes

Maximization

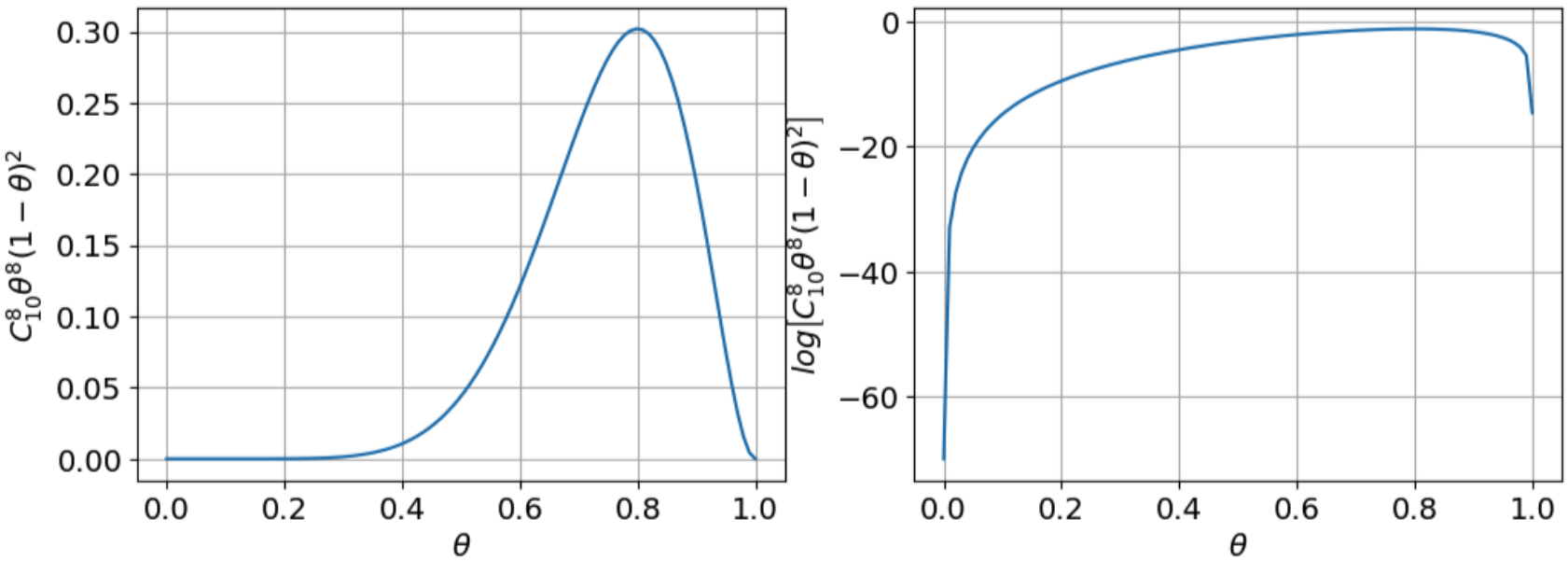
- ceea ce dorim este să maximizăm log-likelihood-ul pentru funcția $P(X|\pi, \theta)$:

$$\log P(X|\pi, \theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\theta_k) \tag{27}$$

- unde θ_k în cazul distribuției normale sunt parametrii săi μ_k și σ_k
- practic media se va obține ca o valoare așteptată a valorilor observate ponderate cu probabilitatea ca $t_k = 1$ când am observat x_n:

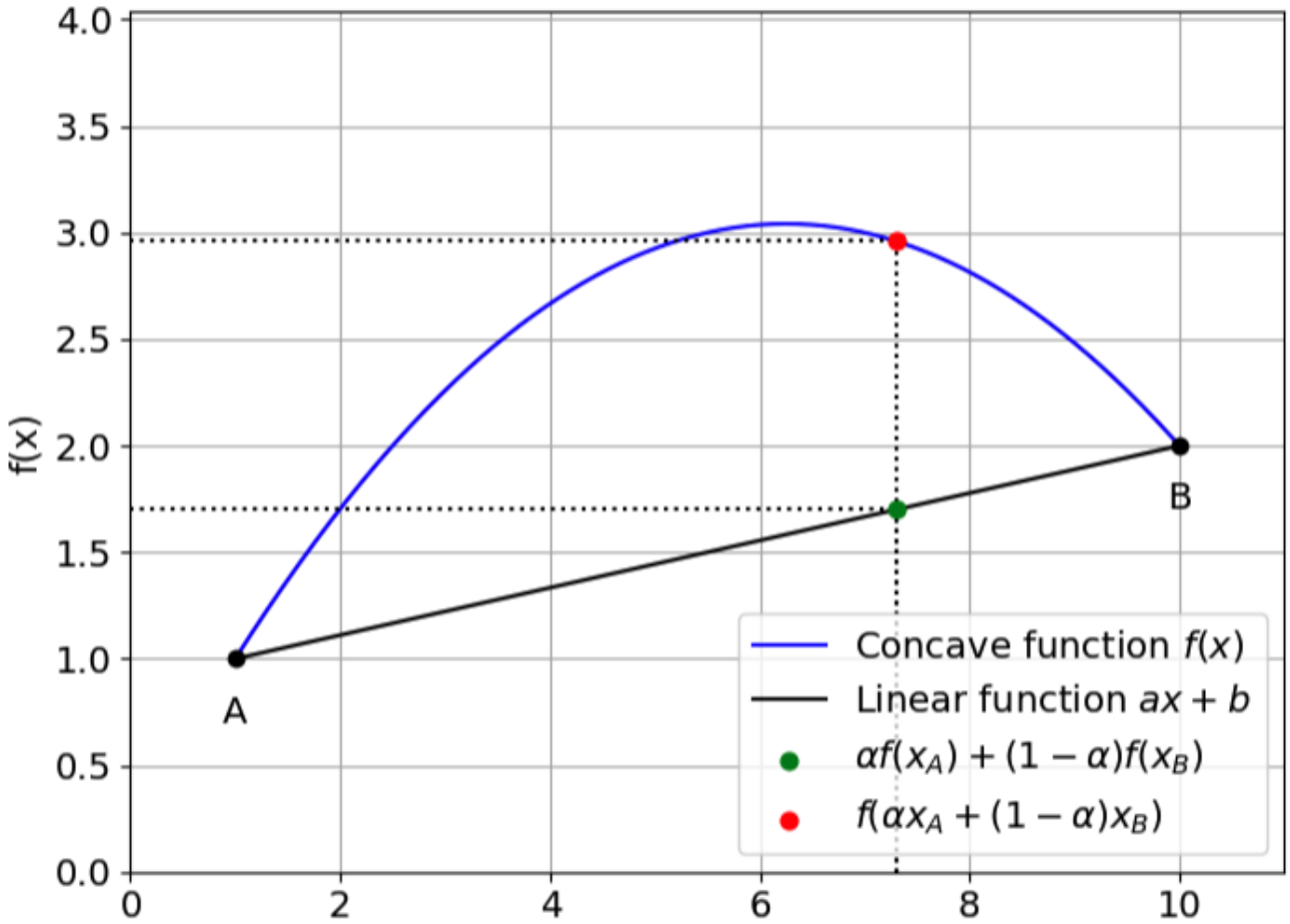
$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(t_{nk}) x_n}{\sum_{n=1}^{N} \gamma(t_{nk})} \tag{28}$$

Motivul de alegere a maximizării funcției log



- funcția densitate de probabilitate (stânga) nu este convexă, spre deosebire de logaritmul său (dreapta)
- maximul nu se schimbă
- pentru funcții convexe avem algoritmi de optimizare care converg (găsesc o soluție), e.g. SGD
- algoritmul EM se pretează pentru optimizare unde nu putem aplica SGD din cauza constrângerilor

Funcție concavă



- o funcție concavă verifică, pentru orice x, $0 \leq \alpha \leq 1$:

$$f(\alpha x + (1 - \alpha)x) \geq \alpha f(x) + (1 - \alpha) f(x) \tag{29}$$

Modele combinate (mixte)

- modelele mixte realizează o asociere "soft" între date și cluster
- mai multe modele se suprapun ca să explice datele generate
- clusterele se pot suprapune, spre deosebire de K-means
- soft-clustering se face folosind probabilitățile
- fiecare cluster corespunde unei distribuții de probabilitate și fiecare punct e văzut ca venind dintr-o astfel de distribuție
- datele sunt:
  - continue - folosim distribuții gaussiene
  - discrete - folosim distribuții multinomiale (zar cu K fețe aruncat de N ori / vs. binomiale)
- fiecare distribuție gaussiană are 2 parametri, media (vector) și matricea de covariație ($d \times d$) care descrie forma gaussienei
- potrivim o distribuție mixtă la date!

Modelul mixt în 1D (1)

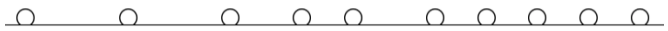
- datele observate $x_1 \dots x_n$ vin din $K = 2$ distribuții gaussiene
- dacă am ști etichetările punctelor, ar fi simplu de estimat parametrii distribuțiilor:

$$\mu_g = \frac{x_1 + x_2 \dots x_{n_g}}{n_g} \tag{30}$$

$$\sigma_g^2 = \frac{(x_1 - \mu_g)^2 + (x_2 - \mu_g)^2 \dots (x_{n_g} - \mu_g)^2}{n_g} \tag{31}$$

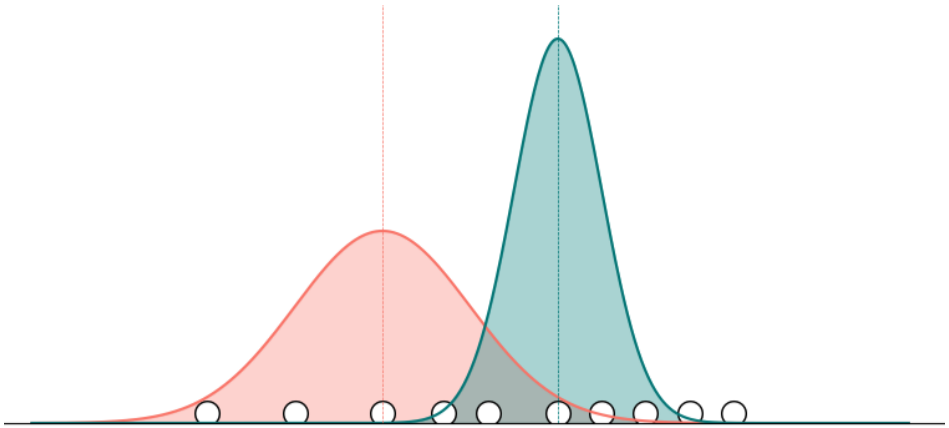
- în realitate nu știm sursa (culoarea), deci nu putem estima parametrii

Modelul mixt în 1D (2)



- dacă am avea cumva mediile și dispersiile, putem calcula probabilitățile cărui cluster le-ar aparține punctele:

$$p(g|x_i) = \frac{p(x_i|g)p(g)}{p(x_i|g)p(g) + p(x_i|r)p(r)} \tag{32}$$

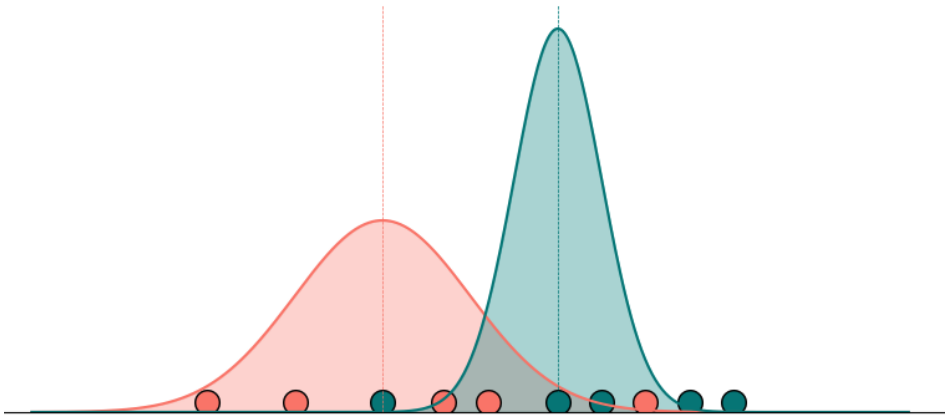


Modelul mixt în 1D (3)

- dacă am avea cumva mediile și dispersiile, putem calcula probabilitățile cărui cluster le-ar aparține punctele (calculul pentru posterior probability):

$$p(g|x_i) = \frac{p(x_i|g)p(g)}{p(x_i|g)p(g) + p(x_i|r)p(r)} \tag{33}$$

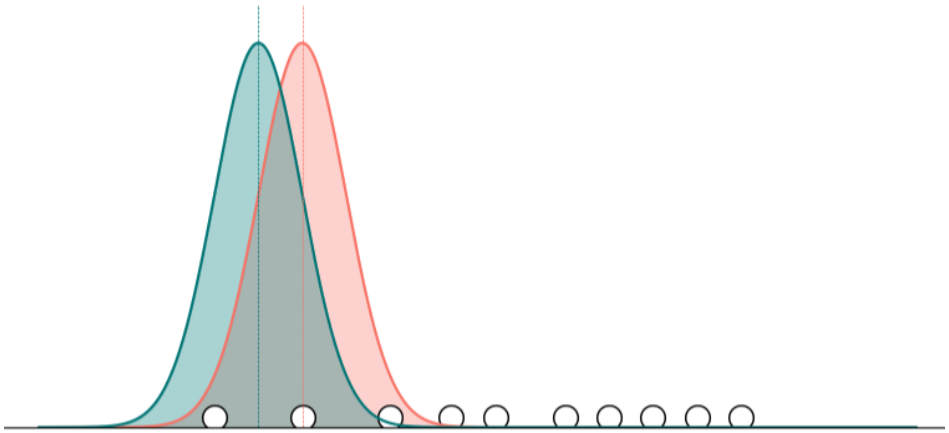
$$p(x_i|g) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left\{-\frac{(x_i - \mu_i)^2}{2\sigma_g^2}\right\} \tag{34}$$



Expectation Maximization (EM)

- chicken and egg problem
  - ne trebuie (μ_g, σ_g^2) și (μ_r, σ_r^2) pentru a presupune sursa (colorarea)
  - ne trebuie etichetarea pentru a calcula parametrii (μ_g, σ_g^2) și (μ_r, σ_r^2)
- algoritmul EM
  - pornește cu gaussiene plasate aleator (μ_g, σ_g^2) și (μ_r, σ_r^2)
  - (E)calculează pentru fiecare punct probabilitatea posterior, $p(g|x_i)$
  - (M) calculează $\mu_g = \frac{\sum_i p(g|x_i) * x_i}{\sum_i p(g|x_i)}$
  - (M) fă un calcul similar pentru σ_g^2
  - (M) reevaluează pobabilitățile prior $p(g) = \frac{\sum_i p(g|x_i)}{n}$
  - repetă pașii până la convergență (estimările parametrilor nu se mai schimbă)

EM (1)

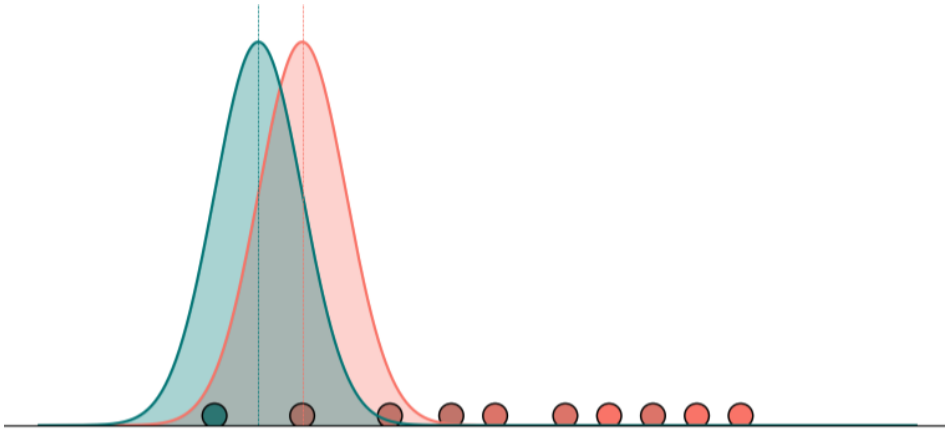


$$p(x_i|g) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left\{-\frac{(x_i - \mu_i)^2}{2\sigma_g^2}\right\}$$

$$g_i = p(g|x_i) = \frac{p(x_i|g)p(g)}{p(x_i|g)p(g) + p(x_i|r)p(r)}$$

$$r_i = 1 - g_i$$

EM (2)



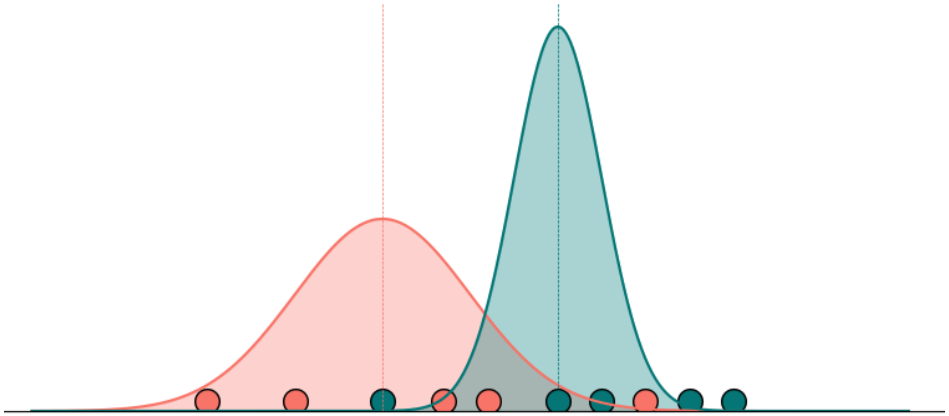
$$p(x_i|g) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left\{-\frac{(x_i-\mu_i)^2}{2\sigma_g^2}\right\}$$

$$g_i = p(g|x_i) = \frac{p(x_i|g)p(g)}{p(x_i|g)p(g) + p(x_i|r)p(r)}$$

$$\mu_g = \frac{g_1x_1 + g_2x_2 + \cdots + g_nx_n}{b_1 + b_2 + \ldots b_n} \text{ and } \mu_r$$

$$\sigma_g^2 = \frac{g_1(x_1-\mu_g)^2 + g_2(x_2-\mu_g)^2 + \cdots + g_n(x_n-\mu_g)^2}{b_1 + b_2 + \ldots b_n} \text{ and } \sigma_r^2$$

EM (3)



- putem reestima și priors:

$$p(g) = \frac{g_1 + g_2 + \dots g_n}{n}$$
$$p(r) = 1 - p(g)$$

Alegerea numărului de clustere K (1)

$$L = logP(x_1 \dots x_n) = \sum_{i=1}^{n} log \sum_{k=1}^{K} P(x_i|k)P(k) \tag{35}$$

- inițializăm K gaussiene în mod random
- K în acest caz este numărul de componente al modelului mixt gaussian (GMM), fiecare componentă asociază o probabilitate lui x_i (cât de probabil e acel punct pentru acea gaussiană înmulțit cu prior)
- $K = n$: fiecare punct are propria sursă de date, dispersia foarte mică (spikes peste data points), likelihood foarte mare, putere de generalizare slabă
- putem introduce un training set și validation set, alegem K cel mai mic; uneori tot ajungem la $K = n$

Alegerea numărului de clustere K (2)

- briciul lui Occam: alegem cel mai mic K care dă o potrivire bună
- Bayes Information Criterion (BIC):

$$\max_p \left\{ L - \frac{1}{2} p \log n \right\} \tag{36}$$

p este numărul de parametri al modelului L mare - cât de bine se potrivește modelul pe date

- Akaike Information Criterion (AIC):

$$\min_p \{2p - L\} \tag{37}$$

- pentru un model mix de 2 gaussiene, sunt 5 parametri (priors)
- în practică GMM sunt folosite pentru a crea features (folosim cele K gaussiene), K va da astfel numărul de features - ajustăm K folosind modelul ce le folosește (e.g. clasificator)

Bibliografie

1. Viktor Lavrenko, EM Algorithm, https://www.youtube.com/watch?v=3JYcCbO5s6M&list=PLBv09BD7ez_7beI0_fuE96lSbsr_8K8YD&index=2
2. H. Hrisov, Intuitive Explanation of the EM Technique, https://www.baeldung.com/cs/expectation-maximization-technique
3. P. Abbeel, Maximum Likelihood (ML), Expectation Maximization (EM), UC Berkeley EECS, https://people.eecs.berkeley.edu/~pabbeel/cs287-fa13/slides/Likelihood_EM_HMM_Kalman.pdf
4. A. Biarnes, Gaussian Mixture Models and Expectation-Maximization (A full explanation), https://towardsdatascience.com/gaussian-mixture-models-and-expectation-maximization-a-fu
5. Bishop, Christopher M., Pattern Recognition and Machine Learning, New York, Springer, 2006, chapter 9