Învăţare automată

— Licenţă, anul III, 2021-2022 —
examen „suplimentar“ II (exerciţii teoretice)

Nume student: **Grupa:**

1. **(Distribuţia gaussiană unidimensională: estimarea mediei în sensul MLE şi respectiv MAP, atunci când varianţa este cunoscută)**

În această problemă ne propunem să calculăm *estimatorul de verosimilitate maximă* (engl., maximum likelihood estimator, MLE) şi *estimatorul de probabilitate maximă a posteriori* (engl., maximum a posteriori probability (MAP) estimator) pentru media unei distribuţii gaussiene unidimensionale. Concret, presupunem că avem n instanţe, $x_1, \ldots, x_n$ generate în mod independent de către o distribuţie normală cu varianţa *cunoscută*, σ^2, şi media *necunoscută*, μ.

Calculaţi estimatorul MLE pentru media μ. Elaboraţi calculele în mod detaliat.

2. **(Distribuţia gaussiană, cazul $\mu = 0$: estimarea varianţei în sensul MLE)**

Fie X o variabilă aleatoare de distribuţie normală (gaussiană) cu media 0 şi varianţa σ^2, adică $X \sim \mathcal{N}(0, \sigma^2)$.

Găsiţi estimarea de verosimilitate maximă (engl., maximum likelihood estimate) pentru parametrul σ^2, adică σ^2_{MLE}.

3. **(Distribuţia gaussiană unidimensională: estimarea varianţei în sensul MLE [în cazul când nu se impun restricţii asupra mediei μ])**

Fie $x_1, \ldots, x_n \in \mathbb{R}$ instanţe generate în mod independent şi identic distribuite conform gaussienei $N(x|\mu, \sigma^2)$.

***Comentarii*:**
1. Vă reamintim că funcţia densitate de probabilitate (p.d.f.) a distribuţiei gaussiene unidimensionale este definită de expresia:

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

2. La problema 1.a am calculat estimatorul MLE al mediei μ pentru această distribuţie. (Am notat acest estimator cu μ_{MLE}.)

Calculaţi estimatorul MLE al varianţei σ^2 (notat cu σ^2_{MLE}), presupunând că nu se impune nicio restricţie asupra lui μ.[1]

[1] Este util de ştiut de asemenea că la problema 2 se cere ca (tot pentru distribuţia gaussiană unidimensională) să se estimeze varianţa σ^2 în sens MLE, presupunând însă că media μ este 0.

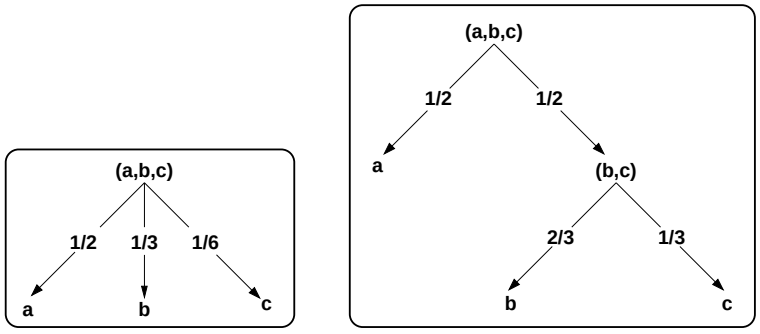
4. **(Redescoperirea definiţiei entropiei pornind de la un set de *proprietăţi dezirabile* ale ei)**

Prin definiţie, *entropia* (în sens Shannon) a unei variabile aleatoare discrete X ale cărei valori sunt luate cu probabilităţile $p_1, p_2, \ldots, p_n$ este $H(X) = -\sum_i p_i \log p_i$. Însă legătura dintre această definiţie formală şi obiectivul avut în vedere — şi anume, acela de a exprima gradul de *incertitudine* cu care se produc valorile unei astfel de variabile aleatoare — nu este foarte intuitivă.

Scopul acestui exerciţiu este de a arăta că orice funcţie $\psi_n(p_1, \ldots, p_n)$ care satisface trei proprietăţi dezirabile (sau, axiome) pentru entropie este în mod necesar de forma $-K \sum_i p_i \log p_i$ unde K este o constantă reală pozitivă. Iată care sunt aceste *proprietăţi*:[2]

- **A1. Funcţia $\psi_n(p_1, \ldots, p_n)$ este continuă în fiecare din argumentele ei şi *simetrică*.**
  Din punct de vedere formal, în acest caz simetria se traduce prin egalitatea $\psi_n(p_1, \ldots, p_i, \ldots, p_j, \ldots, p_n) = \psi_n(p_1, \ldots, p_j, \ldots, p_i, \ldots, p_n)$ pentru orice $i \neq j$. Informal spus, dacă două dintre valorile care sunt luate de variabila aleatoare X (şi anume x_i şi x_j) îşi schimbă între ele probabilităţile (p_i şi respectiv p_j), valoarea entropiei lui X nu se schimbă.
- **A2. Funcţia $\psi_n(1/n, \ldots, 1/n)$ este strict *crescătoare* în raport cu n.**
  Altfel spus, dacă toate evenimentele sunt echiprobabile, atunci entropia creşte odată cu numărul de evenimente posibile.
- **A3. Dacă faptul de a alege între mai multe evenimente posibile poate fi realizat prin mai multe alegeri succesive, atunci $\psi_n(p_1, \ldots, p_n)$ trebuie să se poată scrie ca o sumă ponderată a entropiilor calculate la fiecare stadiu / alegere.**
  De *exemplu*, dacă evenimentele (a, b, c) se produc respectiv cu probabilităţile $(1/2, 1/3, 1/6)$, atunci acest fapt poate fi echivalat cu
  - **a alege mai întâi cu probabilitate de $1/2$ între a şi (b, c),**
  - **urmat de a alege între b şi c cu probabilităţile $2/3$ şi $1/3$ respectiv.**
    (A se vedea imaginile de mai jos, Encoding 1 şi Encoding 2.)

  Din punct de vedere formal, proprietatea A3 impune ca, pe acest exemplu, $\psi_3(1/2, 1/3, 1/6)$ să fie egal cu $\psi_2(1/2, 1/2) + 1/2 \cdot \psi_2(2/3, 1/3)$.



Encoding 1 Encoding 2

Aşadar, în acest exerciţiu vi se cere să arătaţi că dacă o funcţie de n variabile $\psi_n(p_1, \ldots, p_n)$ satisface proprietăţile A1, A2 şi A3 de mai sus, atunci există $K \in \mathbb{R}^+$ astfel încât $\psi_n(p_1, \ldots, p_n) = -K \sum_i p_i \log p_i$ unde $K \in \mathbb{R}_+$ este o constantă.[3]

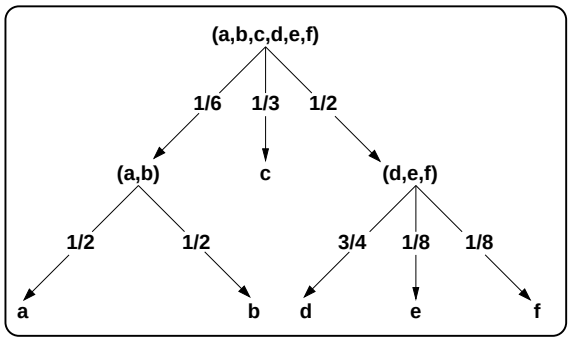
[2] LC: Deşi nu se specifică în enunţul original al problemei, este necesar / natural să considerăm şi proprietatea următoare: [A0.] $\psi_1(1) = 0$ pentru că ψ_n este văzută că măsură a *dezordinii* / *incertitudinii*, iar în cazul particular $n = 1$ nu există niciun fel de dezordine / incertitudine.

[3] În demonstraţie, vom vedea, va rezulta $K = \dfrac{1}{\log s} \psi_s\left(\dfrac{1}{s}, \ldots, \dfrac{1}{s}\right)$ pentru un număr oarecare $s \in \mathbb{N}^* \backslash \{1\}$, fixat.

***Indicaţie*:**
Veţi face rezolvarea acestei probleme în mod gradual, parcurgând următoarele puncte (dintre care primele două puncte au rolul de a vă acomoda cu noţiunile din enunţ):

a. Arătaţi că $H(1/2,1/3,1/6) = H(1/2,1/2)+\frac{1}{2}H(2/3,1/3)$**. Altfel spus, verificaţi faptul că definiţia clasică a entropiei,** $H(X) = \sum_i p_i \log 1/p_i$**, satisface proprietatea A3 pe *exemplul* care a fost dat mai sus.**

b. Calculaţi entropia în cazul distribuţiei / „codificării" din figura alăturată, folosind din nou proprietatea A3.



Următoarele întrebări tratează cazul particular $A(n) \stackrel{not.}{=} \psi_n(1/n,1/n,\ldots,1/n)$**.**

c. Arătaţi că

$$A(s^m) = m\,A(s) \text{ pentru orice } s,m \in \mathbb{N}^*. \tag{1}$$

La punctele $d-g$ **de mai jos, pentru orice număr** $t \in \mathbb{N}^*$ **(fixat), vom considera — pe lângă** $n \in \mathbb{N}^*$**, care a fost de fapt introdus atunci când am spus că aici ne ocupăm de** $A(n)$ **— numerele** $s,m \in \mathbb{N}$**, cu** $s > 1$ **astfel încât**[4]

$$s^m \le t^n \le s^{m+1}. \tag{2}$$

d. Verificaţi că, prin logaritmarea[5] **dublei inegalităţi** (2) **şi apoi prin rearanjare, obţinem** $\frac{m}{n} \le \frac{\log t}{\log s} \le \frac{m}{n} + \frac{1}{n}$ **pentru** $s \neq 1$**, şi deci**

$$\left|\frac{m}{n} - \frac{\log t}{\log s}\right| \le \frac{1}{n}. \tag{3}$$

e. Explicaţi de ce $A(s^m) \le A(t^n) \le A(s^{m+1})$**.**

f. Combinând ultima inegalitate de mai sus cu egalitatea (1)**, avem** $A(s^m) \le A(t^n) \le A(s^{m+1}) \Rightarrow m\,A(s) \le n\,A(t) \le (m+1)\,A(s)$**. Verificaţi că**

$$\left|\frac{m}{n} - \frac{A(t)}{A(s)}\right| \le \frac{1}{n} \text{ pentru } s \neq 1. \tag{4}$$

g. Combinând inegalităţile (3) **şi** (4)**, arătaţi că**

$$\left|\frac{A(t)}{A(s)} - \frac{\log t}{\log s}\right| \le \frac{2}{n} \text{ pentru } s \neq 1 \tag{5}$$

şi, în consecinţă

$$A(t) = K \log t \text{ cu } K > 0 \text{ (din cauza proprietăţii A2)}. \tag{6}$$

[4]LC: În idee, am putea să fixăm $s = 2$ şi apoi să alegem $m \in \mathbb{N}$ (în funcţie de t şi n) astfel încât relaţia (2) să fie satisfăcută. Însă raţionamentul următor nu depinde (în esenţă) de valoarea aleasă (şi fixată) pentru s.

[5]Puteţi folosi ca bază a logaritmului orice număr supra-unitar, arbitrar ales, dar fixat.

h. Arătaţi că rezultatul de mai sus ($A(t) = K \log t \Leftrightarrow \psi_t(1/t, \ldots, 1/t) = Kt\frac{1}{t}\log t$) se generalizează uşor la cazul $\psi_k(p_1, \ldots, p_k)$ cu $p_i \in \mathbb{Q}^+$ pentru $i = 1, \ldots, k$ şi $\sum_{i=1}^{k} p_i = 1$:

$$\psi_k(p_1, \ldots, p_k) = -K \sum_i p_i \log p_i.$$

i. În sfârşit, trataţi şi cazul $p_i \in \mathbb{R}^+,\ i = 1, \ldots, k$, cu $\sum_i p_i = 1$.

5. (Algoritmul Bayes [Naiv] gaussian: aplicare pe date din $\mathbb{R}$)

Presupunem că dispunem de setul de date de antrenament din tabelul alăturat; singurul atribut de intrare (X) ia valori reale, iar atributul de ieşire (Y) este de tip Bernoulli, deci ia două valori, notate cu A şi respectiv B.

X	Y
0	A
2	A
3	B
4	B
5	B
6	B
7	B

a. Pornind de la acest set de date, va trebui mai întâi să învăţaţi *parametrii* clasificatorului Bayes gaussian, prin metoda estimării de verosimilitate maximă (MLE).[6] Centralizaţi rezultatele, completând tabelul următor:

$\mu_A =$	$\sigma_A^2 =$	$P(Y = A) =$
$\mu_B =$	$\sigma_B^2 =$	$P(Y = B) =$

b. Notăm $\alpha = p(X = 2|Y = A)$ şi $\beta = p(X = 2|Y = B)$.

- **Cât este $p(X = 2, Y = A)$ în funcţie de α?**
- **Cât este $p(X = 2, Y = B)$ în funcţie de β?**
- **Cât este $p(X = 2)$ în funcţie de α şi β?**
- **Cât este $p(Y = A|X = 2)$ în funcţie de α şi β?**

c. Cum va clasifica algoritmul Bayes [Naiv] gaussian punctul $X = 2$? Puteţi exprima răspunsul fie în funcţie de α şi β, fie — mai bine! — calculând în prealabil valorile lui α şi β în funcţie de parametrii calculaţi / estimaţi la punctul precedent.

6. (Clasificatorul Bayes [Naiv] gaussian, cazul când se foloseşte un singur atribut de intrare: zone de decizie şi graniţe de decizie; analiza diferitelor *cazuri specifice*)

În acest exerciţiu ne propunem să identificăm *zonele de decizie* şi *graniţele de decizie* determinate de clasificatorul Bayes [Naiv] gaussian (abreviat *GB*), atunci când folosim un singur atribut de intrare X, iar atributul de ieşire, pe care îl vom nota cu Y, este binar, deci poate lua două valori, desemnate în continuare cu A şi B. Vom desemna prin $p_A = p$ *probabilitatea de selecţie* pentru clasa A. Prin urmare,

[6] Vedeţi secţiunea corespunzătoare din capitolul de *Fundamente*, în speţă (pentru acest caz) problemele 1.a şi 3.a.

probabilitatea de selecţie pentru clasa B este $p_B = 1-p$. Vom presupune că $p \in (0,1)$ şi $X|(Y=A) \sim \mathcal{N}(x|\mu_A, \sigma_A^2)$, iar $X|(Y=B) \sim \mathcal{N}(x|\mu_B, \sigma_B^2)$.

a. Arătaţi că *regula de decizie* a acestui clasificator Bayes [Naiv] gaussian

$$\hat{Y}_{GB}(X=x) = A \Leftrightarrow p \cdot \mathcal{N}(x|\mu_A, \sigma_A^2) \geq (1-p) \cdot \mathcal{N}(x|\mu_B, \sigma_B^2) \tag{7}$$

devine echivalentă cu

$$\left(\sigma_A^2 - \sigma_B^2\right)(x - x_1)(x - x_2) \geq 0,$$

unde

$$x_1 = \frac{\sigma_A^2\mu_B - \sigma_B^2\mu_A - \sqrt{\Delta'}}{\sigma_A^2 - \sigma_B^2} \quad \textbf{şi} \quad x_2 = \frac{\sigma_A^2\mu_B - \sigma_B^2\mu_A + \sqrt{\Delta'}}{\sigma_A^2 - \sigma_B^2},$$

cu

$$\Delta' \overset{not.}{=} \sigma_A^2\sigma_B^2 \left[(\mu_A - \mu_B)^2 + (\sigma_A^2 - \sigma_B^2)\ln\left(\frac{1-p}{p} \cdot \frac{\sigma_A}{\sigma_B}\right)^2\right],$$

în *condiţiile* în care $\sigma_A^2 \neq \sigma_B^2$ şi $\Delta' \geq 0$.

b. Arătaţi că atunci când $\sigma_A^2 = \sigma_B^2 \overset{not.}{=} \sigma^2$ şi $\mu_A \neq \mu_B$, regula de decizie (7) este echivalentă cu

$$x \geq x_0 \textbf{ dacă } \mu_A > \mu_B$$

şi, respectiv

$$x \leq x_0 \textbf{ dacă } \mu_A < \mu_B,$$

unde

$$x_0 = \frac{\mu_A + \mu_B}{2} + \frac{\sigma^2}{\mu_A - \mu_B} \cdot \ln\frac{1-p}{p}.$$

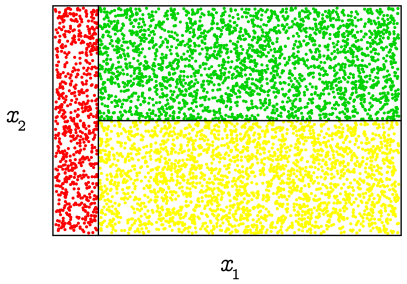
c. Arătaţi că este posibil ca în anumite cazuri să avem $\Delta' < 0$, adică inegalitatea (7) să fie ori adevărată pentru orice $x \in \mathbb{R}$, ori falsă pentru orice $x \in \mathbb{R}$. Altfel spus, arătaţi că pot exista combinaţii de valori pentru parametrii σ_A şi σ_B (ambii strict pozitivi), μ_A şi μ_B din $\mathbb{R}$, precum şi $p \in (0,1)$, astfel încât

$$(\mu_A - \mu_B)^2 + (\sigma_A^2 - \sigma_B^2)\ln\left(\frac{1-p}{p} \cdot \frac{\sigma_A}{\sigma_B}\right)^2 < 0. \tag{8}$$

7. (Arbori de decizie cu variabile continue: ID3 ca algoritm "greedy")

Fie următoarea problemă de clasificare ternară.

Considerăm că în figura alăturată regiunea dreptunghiulară este populată în mod dens cu puncte caracterizate de două atribute numerice (continue), x_1 şi x_2. Cele trei sub-dreptunghiuri (roşu, verde şi galben) reprezintă trei clase de puncte, $C_1, C_2,$ şi C_3.



Dimensiunile $x_1 \times x_2$ ale dreptunghiurilor roşu, verde şi galben sunt 1×6, 7×3 şi respectiv 7×3. Dreptunghiul roşu este populat în mod uniform cu 6000 de puncte din clasa C_1. Dreptunghiul verde este populat în mod uniform cu 42000 de puncte din clasa C_2. Dreptunghiul galben este populat în mod uniform cu 42000 de puncte din clasa C_3. Pentru simplitate, nu vom considera alte puncte decât acestea.

a. Care este numărul minim de noduri de test pe care trebuie să le aibă un arbore de decizie pentru a clasifica în mod corect acest set de date?

b. Câte noduri de test are arborele de decizie obţinut în urma antrenării algoritmului ID3 pe acest set de date, folosind criteriul maximizării câştigului de informaţie?

***Indicaţie*: Pentru a determina entropiile condiţionale minime, puteţi folosi proprietatea A3 de la problema 4 de la capitolul de *Fundamente*.**

c. Avem acelaşi număr de noduri în cele două cazuri de mai sus, sau nu? Care credeţi că este explicaţia?

d. Un arbore de decizie poate să clasifice setul de date din figura de mai sus cu 100% acurateţe (presupunând că nu există zgomote la nivel de etichete). Ce condiţii trebuie să satisfacă în general un set de date de acest gen astfel încât arborele de decizie rezultat în urma antrenării să fie cât mai compact şi să producă o acurateţe de 100%?

***Indicaţie*: Fiecare nod intern al arborelui de decizie corespunde unui test bazat pe o singură trăsătură. Gândiţi-vă ce fel de clase de funcţii / graniţe de separare corespund unui astfel de arbore de decizie.**

8. (O clasă de concepte învăţabile în sens empiric γ-slab cu ajutorul compaşilor de decizie: seturile de instanţe din $\mathbb{R}$, care sunt etichetate în mod consistent)

***Rezumat*:** În această problemă vom arăta că, în ipoteza că instanţele de antrenament cu care lucrăm au un singur atribut, x, care ia valori în $\mathbb{R}$, iar aceste instanţe (x_i, cu $i = 1, \ldots, m$) sunt etichetate în mod „consistent“ (adică, necontradictoriu), există o constantă $\gamma > 0$ care reprezintă o garanţie (engl., guarantee, or edge) pentru învăţabilitate empirică „slabă“, folosind compaşi de decizie în conjuncţie cu algoritmul AdaBoost.[7]

Din punct de vedere *analitic*, compaşii de decizie determinaţi de praguri de separare (engl., thresholding-based decision stumps) pot fi definiţi ca nişte funcţii-prag (sau, funcţii-treaptă), care sunt indexate pe de o parte de un prag $s \in \mathbb{R}$ şi pe de altă parte de un semn $+/-$, în felul următor:

$$\phi_{s,+}(x) = \begin{cases} 1 & \text{dacă } x \geq s \\ -1 & \text{dacă } x < s \end{cases} \quad \text{iar} \quad \phi_{s,-}(x) = \begin{cases} -1 & \text{dacă } x \geq s \\ 1 & \text{dacă } x < s. \end{cases}$$

Prin urmare, $\phi_{s,+}(x) = -\phi_{s,-}(x)$ pentru orice $x \in \mathbb{R}$.

Dat fiind un set de date de antrenament consistent etichetate $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, cu $x_i \in \mathbb{R}$ şi $y_i \in \{-1, +1\}$ pentru $i = 1, \ldots, m$, *obiectivul* nostru este să investigăm existenţa unei *garanţii* pentru învăţabilitate „slabă“ pe setul S. Vom demonstra că există o valoare $\gamma > 0$, astfel încât pentru *orice* distribuţie probabilistă p definită pe S, putem găsi un prag $s \in \mathbb{R}$ cu proprietatea următoare:

$$error_p(\phi_{s,+}) \leq \frac{1}{2} - \gamma \quad \text{sau} \quad error_p(\phi_{s,-}) \leq \frac{1}{2} - \gamma,$$

[7] Noţiunile de învăţabilitate empirică „slabă“ şi garanţie (γ) pentru învăţabilitate empirică „slabă“ au fost definite la problema 23.e. Folosind notaţiile de acolo, vă reamintim că *garanţia* $\gamma \in (0, 1/2)$ are proprietatea că la orice iteraţie t a algoritmului AdaBoost există o ipoteză „slabă“ h_t astfel încât

$$error_{D_t}(h_t) \overset{def.}{=} \sum_{i=1}^{m} D_t(i) \cdot 1_{\{h_t(x_i) \neq y_i\}} \leq \frac{1}{2} - \gamma.$$

În formula aceasta, simbolul $1_{\{\ldots\}}$ desemnează binecunoscuta funcţie indicator. Concret, funcţia $1_{\{h_t(x_i) \neq y_i\}}$ ia valoarea 1 pentru acei i pentru care $h_t(x_i) \neq y_i$, şi 0 în caz contrar. Vă reamintim că p este o distribuţie probabilistă discretă dacă $p_i \geq 0$ pentru $i = 1, \ldots, m$ şi $\sum_{i=1}^{m} p_i = 1$.

unde

$$error_p(\phi_{s,+}) \stackrel{def.}{=} \sum_{i=1}^{m} p_i \cdot 1_{\{y_i \neq \phi_{s,+}(x_i)\}} \quad \textbf{şi} \quad error_p(\phi_{s,-}) \stackrel{def.}{=} \sum_{i=1}^{m} p_i \cdot 1_{\{y_i \neq \phi_{s,-}(x_i)\}}.$$

Facem *presupunerea* că toate instanţele de antrenament sunt distincte, deci nu există $x_i = x_j$ cu $i \neq j$. De asemenea, vom presupune — fără a reduce generalitatea; însă acest fapt ne va permite să simplificăm notaţiile din problemă — că

$$x_1 > x_2 > \ldots > x_m.$$

a. Arătaţi că, dat fiind S, pentru orice prag $s \in \mathbb{R}$ (fixat), dacă notăm cu $m_0(s) \in \{0, 1, \ldots, m\}$ indicele care are proprietatea că $x_i \geq s$ pentru orice $i \leq m_0(s)$ şi $x_i < s$ pentru orice $i > m_0(s)$,[8] atunci au loc egalităţile

$$error_p(\phi_{s,+}) = \frac{1}{2} - \frac{1}{2}\Big(\sum_{i=1}^{m_0(s)} y_i p_i \; - \sum_{i=m_0(s)+1}^{m} y_i p_i \Big)$$

$$error_p(\phi_{s,-}) = \frac{1}{2} - \frac{1}{2}\Big(\sum_{i=m_0(s)+1}^{m} y_i p_i \; - \sum_{i=1}^{m_0(s)} y_i p_i \Big).$$

***Convenţie*: În expresii precum cele de mai sus, veţi trata sumele indexate pe mulţimi vide ca fiind egale cu 0. Aşadar, $\sum_{i=1}^{0} a_i = 0$, oricare ar fi termenii a_i şi, similar, $\sum_{i=m+1}^{m} a_i = 0$.**

b. Arătaţi că, dat fiind setul de date de antrenament S, există o valoare $\gamma > 0$, care poate depinde de m, numărul de instanţe din S (dar nu şi de distribuţiile probabiliste p care pot fi definite pe S), astfel încât pentru orice astfel de distribuţie p, putem găsi un indice m_0 cu proprietatea următoare:

$$|f(m_0)| \geq 2\gamma, \text{ unde } f(m_0) \stackrel{def.}{=} \sum_{i=1}^{m_0} y_i p_i \; - \sum_{i=m_0+1}^{m} y_i p_i.$$

***Sugestie*: Analizaţi diferenţa $f(m_0 + 1) - f(m_0)$.**
Care este valoarea pe care aţi găsit-o pentru γ?

c. Ţinând cont de răspunsurile pe care le-aţi dat la punctele a şi b, cât anume este *garanţia* [pentru învăţabilitate empirică „slabă"] pe care compaşii de decizie o furnizează pentru orice set de date de antrenament $S = \{(x_1, y_1), \ldots (x_m, y_m)\}$, considerând că toate instanţele $x_i \in \mathbb{R}$, cu $i = 1, \ldots, m$ sunt distincte?

d. Puteţi indica o margine superioară (engl., upper bound) pentru numărul de compaşi de decizie necesari pentru a obţine eroare la antrenare 0 pe un astfel de set de date de antrenament (S)?

9. (Algoritmul 1-NN [comparativ cu clasificatorul Bayes Optimal]: o margine superioară pentru *eroarea medie asimptotică* [la antrenare])

Un rezultat interesant obţinut de Cover şi Hart (1967) arată că, atunci când numărul datelor de antrenament tinde la infinit, iar datele de antrenament umplu spaţiul în mod dens, rata medie a erorii produsă de către clasificatorul 1-NN este mărginită

[8] Ştim că un astfel de indice există findcă $x_1 > x_2 > \ldots > x_m$.

superior de dublul ratei medii a erorii pentru clasificatorul Bayes Comun (engl., Joint Bayes), care este numit adeseori şi *Bayes Optimal* (engl., Optimal Bayes).

În acest exerciţiu vi se va arăta, pas cu pas, cum se demonstrează rezultatul lui Cover şi Hart în cazul particular al clasificării binare. Aşadar, fie $x_1, x_2, \ldots$ instanţele de antrenament, iar $y_1, y_2, \ldots$ etichetele corespunzătoare, cu $y_i \in \{0, 1\}$. Putem considera instanţele x_i ca fiind puncte într-un spaţiu euclidian d-dimensional.

Notăm $p_y(x) = P(X = x \mid Y = y)$ probabilitatea condiţionată care reprezintă distribuţia instanţelor din clasa y. Vom presupune că aceste probabilităţi condiţionate sunt continue în raport cu variabila x şi că $p_y(x) \in (0, 1)$ pentru orice x şi orice y. Notăm cu θ probabilitatea ca un exemplu de antrenament selectat în mod aleatoriu să fie din clasa 1, aşadar $\theta \stackrel{not.}{=} P(Y = 1)$. Din nou, presupunem că $\theta \in (0, 1)$.

a. Calculaţi probabilitatea ca o instanţă oarecare x să aparţină clasei 1: $q(x) \stackrel{not.}{=} P(Y = 1 \mid X = x)$. Exprimaţi $q(x)$ în funcţie de $p_0(x)$, $p_1(x)$ şi θ.

b. Clasificatorul Bayes Optimal asignează unui punct dat x cea mai probabilă clasă, $\arg\max_y P(Y = y \mid X = x)$. (Aceasta implică faptul că algoritmul Bayes Optimal maximizează probabilitatea clasificării corecte a tuturor datelor.) Considerând o instanţă oarecare x, calculaţi probabilitatea ca x să fie clasificat greşit folosind clasificatorul Bayes Optimal, în funcţie de probabilitatea $q(x) \stackrel{not.}{=} P(Y = 1|X = x)$ care tocmai a fost calculată la punctul precedent. Veţi desemna această nouă probabilitate cu $Error_{Bayes}(x)$.

c. Acum considerăm clasificatorul 1-NN. Acesta îi asignează unei instanţe oarecare de test x eticheta celei mai apropiate instanţe de antrenament x'. Dată fiind o instanţă de antrenament x (aleasă în mod arbitrar, dar fixată), calculaţi *eroarea „aşteptată“* (engl., expected error) produsă de către clasificatorul 1-NN, adică probabilitatea ca instanţa x să fie clasificată greşit. Notaţi această probabilitate cu $Error_{1\text{-}NN}(x)$ şi exprimaţi-o sub forma unei funcţii definită în raport cu probabilităţile $q(x)$ şi $q(x')$.

d. În *cazul asimptotic*, numărul de exemple de antrenament al fiecărei clase tinde la infinit, iar datele de antrenament umplu spaţiul în mod dens. Atunci $q(x') \to q(x)$, unde, ca şi mai sus, x' este cel mai apropiat vecin al lui x.[9] Făcând această substituţie în rezultatul obţinut la punctul anterior, deduceţi expresia *erorii asimptotice* pentru clasificatorul 1-NN în punctul x, adică $\lim_{x' \to x} Error_{1\text{-}NN}(x)$, în funcţie de probabilitatea $q(x)$.

e. Arătaţi că eroarea asimptotică obţinută la punctul d este mai mică decât dublul erorii clasificatorului Bayes Optimal obţinută la punctul b, adică:

$$\lim_{x' \to x} Error_{1\text{-}NN}(x) \leq 2 Error_{Bayes}(x).$$

În final, din această inegalitate deduceţi relaţia corespunzătoare între ratele medii ale erorilor:[10]

$$E[\lim_{n \to \infty} Error_{1\text{-}NN}] \leq 2E[Error_{Bayes}].$$

---

[9] Adică, $P(Y = 1|X = x') \to P(Y = 1|X = x)$. Aceasta se justifică ţinând cont de continuitatea lui $p_y(x) \stackrel{not.}{=} P(X = x|Y = y)$ care a fost asumată în enunţ şi, de asemenea, de rezultatul obţinut la punctul a.

[10] Cititorul atent va remarca faptul că în expresia de mai jos ($E[\lim_{n \to \infty} Error_{1\text{-}NN}]$) s-a înlocuit $\lim_{x \to x'}$ (folosită anterior) cu $\lim_{n \to \infty}$, pentru că se face trecerea la medii. Când $n \to \infty$, conform presupoziţiilor din enunţ, rezultă $x \to x'$ pentru orice x.

10. **(Compararea clasificatorilor 1-NN şi Bayes Optimal: o margine superioară mai bună pentru *rata medie a erorii asimptotice* a lui 1-NN)**

La problema 9 am demonstrat că *rata medie a erorii* clasificatorului 1-NN este mărginită asimptotic[11] de dublul ratei medii a erorii clasificatorului Bayes Optimal.

Arătaţi că — în aceleaşi condiţii ca la problema 9 — se poate obţine o margine chiar mai bună:

$$E[\lim_{n\to\infty} Error_{1\text{-}NN}] \leq 2E[Error_{Bayes}](1 - E[Error_{Bayes}]).$$

11. **(Algoritmul EM: chestiuni calitative / metodologice privind aplicarea în cazul unei mixturi de două distribuţii gaussiene unidimensionale presupunând probabilităţile de selecţie egale şi fixate)**

A fost odată, cu mult timp în urmă, un sat care era situat într-o regiune cu sute de lacuri. În acele lacuri trăiau două specii de peşti, însă în fiecare lac, nu erau decât peşti dintr-o singură specie. (Peştii din lacuri diferite puteau fi de specii diferite.) Peştii din aceste două specii arătau identic, miroseau identic atunci când erau gătiţi şi aveau exact acelaşi gust delicios, însă una dintre specii era otrăvitoare şi oricine mânca peşti din specia respectivă murea. Singura diferenţă observabilă la aceste specii de peşti consta în efectul asupra nivelului pH-ului (aciditatea) apei din lacul în care trăiau. Lacurile cu peşti neotrăvitori aveau pH-ul distribuit conform unei distribuţii gaussiene de medie (μ_{sigur}) şi varianţă (σ^2_{sigur}) necunoscute, iar pH-ul din lacurile cu peşti otrăvitori era distribuit conform unei alte distribuţii gaussiene de medie (μ_{mortal}) şi varianţă (σ^2_{mortal}) de asemenea necunoscute. (Peştii otrăvitori cauzau o aciditate care era în general mai mare / ridicată decât în cazul celorlalţi peşti.)

Aşa cum era firesc, pentru a ieşi din încurcătură — adică pentru a determina caracterul otrăvitor ori neotrăvitor al peştilor din fiecare lac (sau dintr-un lac oarecare, din care încă nu s-au consumat peşti) în funcţie de nivelul pH-ului apei —, sătenii au apelat la învăţarea automată. Cu toate acestea, au avut loc dezbateri aprinse cu privire la modalitatea corectă de aplicare a algoritmului Expectation-Maximization la problema lor.

Pentru fiecare dintre modalităţile prezentate mai jos, indicaţi dacă ea constituie o aplicare corectă a algoritmului EM şi dacă va conduce la o estimare rezonabilă pentru parametrii μ şi σ^2 ai fiecărei clase.

a. Se ghicesc valorile iniţiale pentru parametrii μ şi σ^2 corespunzători fiecărei specii. (1) Pentru fiecare lac, se determină — folosind teorema lui Bayes şi distribuţiile gaussiene de parametri μ şi σ^2 — care este cea mai probabilă specie de peşti asociată lacului respectiv. (2) Se recalculează valorile pentru μ şi σ^2 utilizând metoda estimării de verosimilitate maximă. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

b. Pentru fiecare lac, se ghiceşte probabilitatea (iniţială) că lacul respectiv ar fi populat cu peşti neotrăvitori. (1) Folosind aceste probabilităţi, se estimează în sensul verosimilităţii maxime valorile μ şi σ corespunzătoare fiecărei clase. (2) Utilizând aceste estimări pentru μ şi σ, se recalculează probabilităţile de 'siguranţă' pentru fiecare lac. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

[11] Adică, atunci când $n \to \infty$, unde n este numărul de instanţe de antrenament.

c. Se calculează media şi varianţa nivelului de pH pentru toată mulţimea lacurilor. Se folosesc aceste valori pentru a iniţializa parametrii μ şi σ^2 corespunzători fiecărei specii de peşti. (1) Folosind aceste valori pentru μ şi σ^2, se calculează pentru fiecare lac probabilitatea să conţină peşti otrăvitori. (2) Se găsesc valorile de verosimilitate maximă pentru μ şi σ^2. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

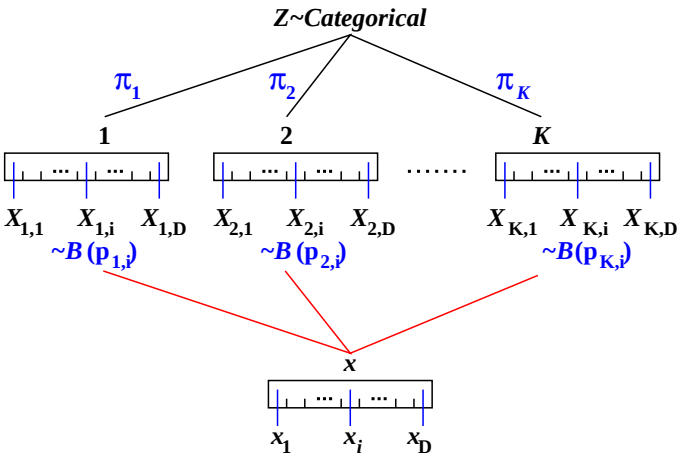
12. (Algoritmul EM pentru rezolvarea unei mixturi de vectori de variabile Bernoulli [independente]; aplicare la recunoşterea cifrelor scrise de mână)

În acest exerciţiu, veţi crea un algoritm de tip EM (expectation-maximization) care să clusterizeze imagini alb-negru. Input-urile $x^{(i)}$ pot fi văzute ca vectori de valori binare corespunzătoare culorilor alb şi negru. Obiectivul este să clusterizăm aceste imagini în [mai multe] grupuri. Pentru a rezolva această problemă, veţi folosi un model de tip mixtură de distribuţii Bernoulli.

A. Mixtura de distribuţii Bernoulli

a. Considerăm un vector de variabile aleatoare binare, $x \in \{0,1\}^D$. Presupunem că fiecare variabilă x_d urmează o distribuţie $Bernoulli(p_d)$, deci $P(x_d = 1) = p_d$. Fie $p \in (0,1)^D$ vectorul rezultat de parametri Bernoulli. Scrieţi expresia probabilităţii $P(x|p)$, considerând că variabilele x_d sunt independente între ele.

b. Să presupunem acum că avem o mixtură de K vectori de distribuţii Bernoulli: fiecare vector $x^{(i)}$ este generat folosind un vector de variabile Bernoulli independente, de parametru $p^{(k)} \overset{not.}{=} (p_1^{(k)}, \ldots, p_D^{(k)})$.



Presupunem că dispunem de o distribuţie aleatoare $\pi \overset{not.}{=} (\pi_1, \ldots, \pi_K)$, cu π_k indicând probabilitatea de selecţie a setului de parametri Bernoulli $p^{(k)}$, pentru $k = 1, \ldots, K$.

Scrieţi expresia probabilităţii $P(x^{(i)}|p, \pi)$, unde cu p notăm (de acum încolo) ansamblul $(p^{(1)}, \ldots, p^{(K)})$.

c. Presupunem că avem input-urile $X = \{x^{(i)}\}_{i=1,\ldots,n}$. Folosind rezultatele de la punctele anterioare, scrieţi expresia log-verosimilităţii datelor X, $\ln P(X|\pi, p)$.

B. Pasul de estimare (engl., expectation step)

d. Acum vom introduce variabilele latente pentru algoritmul EM. Fie $z^{(i)} \in \{0,1\}^K$ un vector indicator, astfel încât $z_k^{(i)} = 1$ dacă vectorul $x^{(i)}$ a fost generat de vectorul de distribuţii $Bernoulli(p^{(k)})$, şi 0 în caz contrar. Fie $Z = \{z^{(i)}\}_{i=1,\ldots,n}$. Cât este probabilitatea $P(z^{(i)}|\pi)$? Dar $P(x^{(i)}|z^{(i)}, p, \pi)$?

***Indicaţie*: Folosiţi artificiul ridicării la putere (engl., exponentiation trick), ţinând cont de faptul că $z_k^{(i)} \in \{0,1\}$.**

e. Folosind cele două probabilităţi calculate la punctul d, scrieţi expresia verosimilităţii datelor complete, $P(X, Z|\pi, p)$.

f. Fie $\mu(z_k) \overset{not.}{=} E[z_k^{(i)}|x^{(i)}, \pi, p]$. Demonstraţi că

$$\mu(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1-x_d^{(i)}}}{\sum_{j=1}^{K} \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1-x_d^{(i)}}}.$$

Fie $\bar{p}$ şi $\bar{\pi}$ noile valori ale parametrilor pe care dorim să le obţinem prin maximizare, iar p şi π valorile lor de la iteraţia precedentă. Folosind aceste notaţii, arătaţi că funcţia „auxiliară", care este necesară pentru pasul M al algoritmului de estimare-maximizare, are expresia următoare:

$$\begin{aligned} & E[\ln P(X, Z|\bar{p}, \bar{\pi})|X, p, \pi] \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \mu(z_k^{(i)}) \left[\ln \bar{\pi}_k + \sum_{d=1}^{D} \left(x_d^{(i)} \ln \bar{p}_d^{(k)} + (1 - x_d^{(i)}) \ln(1 - \bar{p}_d^{(k)}) \right) \right]. \end{aligned}$$

C. Pasul de maximizare

g. Acum trebuie să maximizăm expresia funcţiei „auxiliare" care a fost dedusă la punctul f, în raport cu $\bar{\pi}$ şi $\bar{p}$. Mai întâi, arătaţi că valoarea $\bar{p}$ care maximizează funcţia „auxiliară" este

$$\bar{p}^{(k)} = \frac{\sum_{i=1}^{N} \mu(z_k^{(i)})\, x^{(i)}}{N_k},$$

unde $N_k = \sum_{i=1}^{N} \mu(z_k^{(i)})$.

h. Demonstraţi că valoarea $\bar{\pi}$ care maximizează funcţia „auxiliară" este:

$$\bar{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}.$$

***Sugestie*: Dat fiind faptul că este vorba de rezolvarea unei probleme de optimizare cu restricţii, puteţi apela la metoda multiplicatorilor Lagrange.**

13. (EM pentru combinarea unei clasificări de tip Bayes Naiv cu o mixtură de vectori de distribuţii Bernoulli, folosind şi un termen de penalizare / regularizare. Aplicaţie: recunoaşterea cifrelor scrise de mână)

● ○ ***U. Toronto, Radford Neal, "Statistical Methods for Machine Learning and Data Mining" course, 2014 spring, HW2***

***Obiectivul* dumneavoastră în această problemă va fi să clasificaţi cifre scrise de mână, cu ajutorul unor modele de mixturi [de distribuţii probabiliste] care vor fi obţinute prin maximimzarea verosimilităţii datelor însoţită de un termen de penalizare, folosind algoritmul EM.**

Datele pe care le veţi folosi constau din 800 de imagini de antrenament şi 1000 de imagini de test pentru cifre scrise de mână (din codurile poştale de pe plicuri din SUA). Aceste imagini au fost extrase din bine-cunoscutul set de date MNIST, prin selectare aleatorie dintr-un total de 60000 de imagini de antrenament furnizate, reducând rezoluţia acestor imagini de la 28×28 de pixeli la 14×14 de pixeli, făcând media valorilor pixelilor pentru fiecare bloc de pixeli de

dimensiune 2×2, iar apoi folosind nişte praguri pentru aceste medii în aşa fel încât să obţinem valori binare. Pe pagina web a acestei culegeri[12] este furnizat un fişier de date de antrenament având 800 de linii, fiecare linie conţinând 196 de valori pentru pixeli şi anume, fie 0, fie 1. Tot acolo este furnizat şi un alt fişier care conţine etichetele corespunzătoare acestor 800 de cifre (de la 0 to 9). În mod similar, vă punem la dispoıţie şi un fişier cu 1000 de imagini de test, precum şi un fişier conţinând etichetele acestor 1000 de imagini de test. (Nu veţi avea voie să vă uitaţi la etichetele acestor imagini de test decât la sfârşit de tot, şi anume atunci când va trebui să analizaţi cât de bine se comportă algoritmii de învăţare automată.)

Veţi clasifica aceste imagini pentru cifre scrise de mână folosind un *model generativ*, din care, dată fiind imaginea unei astfel de cifre, veţi deriva probabilităţile pentru cele 10 clase posibile. *Clasa asociată unei cifre de test* este cea care are probabilitatea maximă.

Modelul generativ pe care îl vom folosi estimează *probabilităţile claselor* ca [fiind] *frecvenţele* lor din setul de date de antrenament (ceea ce va determina o distribuţie aproximiativ uniformă — însă nu exact uniformă — peste cele 10 cifre). De asemenea, acest model estimează *distribuţiile de probabilitate ale imaginilor din cadrul fiecărei clase* cu ajutorul unui *model de mixtură cu K componente*, fiecare componentă modelând valorile celor 196 de pixeli ca [fiind] variabile aleatoare *independente*. Va fi convenabil să combinăm toate aceste 10 modele de mixturi [de distribuţii probabiliste] într-un singur *model de mixtură cu $10K$ componente*, care modelează atât valorile pixelilor cât şi etichetele claselor. Totuşi, *probabilităţile* [*condiţionate ale*] *componentelor* din cadrul modelului fiecărei clase vor fi *fixate*, astfel încât K componente vor atribui probabilitatea condiţionată 1 cifrei 0, alte K componente vor atribui probabilitatea condiţionată 1 cifrei 1, alte K vor atribui probabilitatea condiţionată 1 cifrei 2 ş.a.m.d.

Aşadar, modelul probabilist pentru *distribuţia comună* asupra etichetelor y_i, precum şi asupra valorilor pixelilor $x_{i,1}, \ldots, x_{i,196}$ pentru $i = 0, \ldots, 9$ poate fi exprimat sub forma următoare:

$$P(y_i, x_i | \pi, \theta) = \sum_{k=1}^{10K} \left(\pi_k q_{k,y_i} \prod_{j=1}^{196} \theta_{k,j}^{x_{i,j}} (1 - \theta_{k,j})^{1 - x_{i,j}} \right).$$

Instanţele etichetate (x_i, y_i) se presupun a fi independente.

***Parametrii* acestui model sunt *probabilităţile de mixare / selecţie*, $\pi_1, \ldots, \pi_{10K}$, precum şi probabilităţile $\theta_{k,j}$ asociate pixelilor (mai precis, pentru ca aceştia să fie setaţi la valoarea 1), pentru fiecare componentă în parte, adică pentru $k = 1, \ldots, 10K$ şi $j = 1, \ldots, 196$. Probabilităţile condiţionate $q_{k,y}$ ale componentelor fiecărei clase sunt fixate astfel:**

$$q_{k,y} = \begin{cases} 1 & \text{atunci când } k \in \{Ky + 1, \ldots, Ky + K\} \\ 0 & \text{în caz contrar,} \end{cases}$$

pentru $k = 1, \ldots, 10K$ şi $y = 0, \ldots, 9$.

La sfârşitul acestui exerciţiu veţi implementa în Python / R / Matlab o funcţie care identifică valorile parametrilor care maximizează log-verosimilitatea datelor de antrenament plus un *termen de penalizare* (engl., penalty term).[13]

Algoritmul EM poate fi uşor adaptat pentru a găsi [în locul estimării verosimilităţii maxime] estimarea verosimilitatăţii maxime penalizate. În raport cu versiunea generală a algoritmului EM [care a fost prezentată la curs], pasul E se păstrează

[12] Vedeţi https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/UToronto.2014s.RNeal.HW2.EM-for-BernoulliMM-using-the-NBayes-assumption.handwritten-char-reco.data+R-code+sol/

[13] Cu cât valorile acestui termen de penalizare vor fi mai mari, cu atât va fi mai bine.

neschimbat, însă pasul M va maximiza acum o funcţie care reprezintă media log-verosimilităţii penalizate a datelor complete, $E_Q[\ln P(x, z|\theta) + G(\theta)]$, unde $G(\theta)$ este termenul de penalizare.

Scopul penalizării **este acela de a evita ca estimările probabilităţilor asociate pixelilor să fie 0 sau aproape de 0, situaţii care pot cauza probleme la clasificarea instanţelor de test. (Este imediat că dacă o astfel de probabilitate este 0, rezultă că probabilitatea oricărei instanţe de test va fi 0 pentru clasa respectivă.) Termenul de penalizare care se sumează la log-verosimilitate trebuie să fie**

$$G(\theta) = \alpha \sum_{k=1}^{10K} \sum_{j=1}^{196} [\log(\theta_{k,j}) + \log(1 - \theta_{k,j})],$$

unde constanta α controlează „magnitudinea“ penalizării.[14]

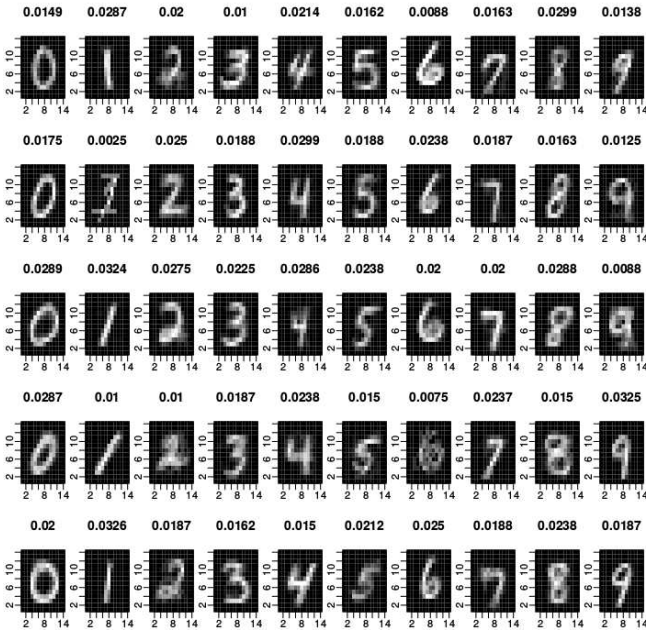
a. Demonstraţi că într-adevăr [regula de actualizare de la] pasul E rămâne la fel ca în cazul versiunii generale a algoritmului EM.

b. La pasul M, reestimarea probabilităţilor de selecţie π va rămâne de asemenea neschimbată [în raport cu cazul general],[15] **însă pentru reestimarea parametrilor θ va trebui să luaţi în considerare şi termenul de penalizare. Demonstraţi că formula pentru actualizarea parametrilor $\theta_{k,j}$ la pasul M este**

$$\hat{\theta}_{k,j} = \frac{\alpha + \sum_{i=1}^{n} r_{i,k} x_{i,j}}{2\alpha + \sum_{i=1}^{n} r_{i,k}},$$

unde $r_{i,k}$ este probabilitatea ca instanţa i să fi fost generată de către componenta k a mixturii, probabilitate care a fost estimată la pasul E.[16]

Comentariu**: Pentru $K = 5$, putem ilustra sub forma unor imagini valorile parametrilor θ care au fost învăţate pentru cele 50 de componente ale mixturii:**



[14] Pentru acest exerciţiu, α va putea fi fixat la valoarea 0.05, deşi în aplicaţii reale el va trebui probabil să fie determinat folosind, de exemplu, metoda cross-validării.

[15] Va trebui să vă convingeţi singuri de acest fapt.

[16] Ca să obţineţi această formulă, aţi putea să porniţi de la demonstraţia care a fost prezentată la problema 12, modificând-o în aşa fel încât să includă şi termenul de penalizare.

Este clar că în general cele cinci componente corespunzătoare fiecărei cifre au identificat diferite variante rezonabile de caligrafiere a cifrelor, exceptând probabil câteva cazuri, cărora le corespund probabilităţi de selecţie destul de mici (indicate sub formă numerică, deasupra imaginilor), aşa cum este caracterul "1" de pe cel de-al doilea rând de imagini.