20. Grupiranje II

Strojno učenje 1, UNIZG FER, ak. god. 2022./2023.

Jan Šnajder, vježbe, v2.2

1 Zadatci za učenje

1. [***Svrha: Razumjeti model miješane gustoće i razlog zašto maksimizacija log-izglednost nije analitički rješiva. Razumjeti kako uvođenje latentnih varijabli rješava taj problem. Razumjeti, na općenitoj razini, E-korak i M-korak. Razumjeti rad algoritma kao maksimizacije log-izglednosti i razumjeti kako ishod ovisi o broju grupa i početnoj inicijalizaciji.***] Algoritam maksimizacije očekivanja (EM-algoritam), kada se koristi za grupiranje, zapravo je poopćenje algoritma K-sredina.

   (a) Što je prednost, a što nedostatak, algoritma maksimizacije očekivanja primijenjenog na GMM u odnosu na algoritam K-sredina?

   (b) Napišite izraz za gustoću $p(\mathbf{x})$ za model miješane gustoće (bez latentnih varijabli) i izraz za pripadnu (nepotpunu) log-izglednost.

   (c) Napišite izraz za mješavinu s latentnim varijablama i izvedite izraz za (potpunu) log-izglednost tog modela. Možemo li dalje raditi izravno s tom log-izglednošću? Zašto?

   (d) Definirajte E-korak i M-korak algoritma maksimizacije očekivanja primijenjenog na Gaussovu mješavinu.

   (e) Skicirajte vrijednost log-izglednosti $\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ modela Gaussove mješavine kao funkcije broja iteracija, i to za tri različite vrijednosti parametra K (broj grupa): $K = 1$, $K = 10$ i $K = 100$. Na istom grafikonu skicirajte krivulju za $K = 10$ kada se za inicijalizaciju središta koristi algoritam K-sredina.

2. [***Svrha: Isprobati rad algoritma hijerarhijskog aglomerativnog grupiranja (HAC) na konkretnom primjeru, za slučaj kada primjeri nisu vektori. Uočiti razliku između udaljenosti i sličnosti te razliku između jednostruke i potpune povezanosti.***] Jednako kao i algoritam K-medoida, algoritam hijerarhijskog algomerativnog grupiranja može se primijeniti u slučajevima kada primjeri nisu prikazani kao vektori značajki te kada umjesto mjere udaljenosti između vektora raspolažemo općenitjom mjerom sličnosti (ili različitosti). Neka je *sličnost* primjera iz $\mathcal{D}$ definirana sljedećom matricom sličnosti:

$$
S = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccc} \quad a & \quad b & \quad c & \quad d & \quad e \end{array} \\
\begin{pmatrix}
1.00 & 0.26 & 0.15 & 0.20 & 0.17 \\
0.26 & 1.00 & 0.24 & 0.31 & 0.31 \\
0.15 & 0.24 & 1.00 & 0.20 & 0.50 \\
0.20 & 0.31 & 0.20 & 1.00 & 0.24 \\
0.17 & 0.31 & 0.50 & 0.24 & 1.00
\end{pmatrix}
\end{array}
$$

   (a) Izgradite dendrogram uporabom jednostrukog povezivanja. Kada bi bilo potrebno napraviti particiju grupa, na kojoj biste razini presjekli taj dendrogram?

   (b) Izgradite dendrogram uporabom potpunog povezivanja. Kada bi bilo potrebno napraviti particiju grupa, na kojoj biste razini presljekli taj dendrogram?

3. [***Svrha: Razumjeti kako se unutarnji kriterij algoritma grupiranja može (pokušati) upotrijebiti za provjeru grupiranja (odabir optimalnog broja grupa). Razumjeti da Akaikeov kriterij u stvari oponaša regulariziranu funkciju pogreške, koja pak aproksimira pogrešku generalizacije.***]

   (a) Skicirajte krivulju log-izglednosti kod EM-algoritma kao funkciju broja grupa K. Obrazložite izgled krivulje. Možete li temeljem ove krivulje odrediti optimalan broj grupa? Kako?

(b) Optimizacija broja grupa K može se provesti nekim kriterijem koji kombinira funkciju pogreške (odnosno log-izglednost) i složenost modela. Takav kriterij odgovara strukturnome riziku modela, koji je minimalan za optimalan broj grupa. Jedan takav kriterij jest Akaikeov informacijski kriterij (AIC):

$$K^* = \underset{K}{\operatorname{argmin}} \big(-2 \ln \mathcal{L}(K) + 2q(K) \big)$$

gdje je $-\ln \mathcal{L}(K)$ negativna log-izglednost podataka za K grupa, a $q(K)$ je broj parametara modela s K grupa.

Pretpostavite da podatci $\mathcal{D}$ u stvarnosti dolaze iz $K = 5$ grupa. Podatke grupiramo dvjema varijantama EM-algoritma: standardni algoritam i preinačeni algoritam s dijeljenom kovarijacijskom matricom (zajednička kovarijacijska matrica procijenjena nad čitavim skupom primjera $\mathcal{D}$ na početku izvođenja algoritma). Skicirajte za ta dva algoritma funkciju koju minimizira Akaikeov minimizacijski kriterij.

2 Zadatci s ispita

1. (T) Algoritam GMM, odnosno model Gaussove mješavine s algoritmom maksimizacije očekivanja kao optimizacijskim postupkom, poopćenje je algoritma K-sredina. **Uz koje uvjete algoritam GMM degenerira u algoritam K-sredina?**

   A Umjesto maksimizacije log-izglednosti, minimizira se negativna log-izglednost, a početna središta se odabiru algoritmom K-sredina

   B Koeficijenti mješavine su jednaki za sve komponente Gaussove mješavine, a kovarijacijske matrice su dijagonalne

   C Kovarijacijska matrica komponenti Gaussove mješavine je jedinična matrica, a maksimizira se negativna log-izglednost

   D Kovarijacijska matrica komponenti Gaussove mješavine je dijeljena i izotropna, a odgovornosti su zaokružene na cijeli broj

2. (T) Algoritam maksimizacije očekivanja (EM-algoritam) maksimizira očekivanje potpune log-izglednosti, što se pokazuje da dovodi i do maksimizacije nepotpune log-izglednosti. **Koja je razlika između potpune i nepotpune log-izglednosti, i zašto maksimiziramo očekivanje potpune log-izglednosti umjesto izravno log-izglednost?**

   A Potpuna log-izglednost je izglednost izračunata na svim primjerima iz neoznačenog skupa primjera, dok je nepotpuna log-izglednost izračunata samo za označene primjere koji se koriste za evaluaciju modela, a očekivanje računamo zato jer je postupak grupiranja stohastičan

   B Potpuna log-izglednost je log-izglednost s neopaženim varijablama, a u slučaju GMM-a to su centroidi i kovarijacijske matrice komponenata, koje procjenjujemo metodom MLE, koja maksimizira očekivanje log-izglednosti

   C Potpuna log-izglednost je log-izglednost modela GMM s latentnim varijablama, koje definiraju koji primjer pripada kojoj grupi, međutim kako to zapravo ne znamo, moramo računati s očekivanjem tih varijabli

   D Potpuna log-izglednost računa se za označene primjere a nepotpuna log-izglednost za neoznačene primjere, a u oba slučaja kod modela GMM računamo očekivanje log-izglednosti jer postupak za različite početne centroide može dati različite log-izglednosti

3. (T) Za procjenu parametara modela GMM tipično se koristi algoritam maksimizacije očekivanja (EM-algoritam). To je iterativan optimizacijski algoritam. **Pod kojim uvjetima EM-algoritam**

(primijenjen na model GMM) konvergira, i kamo?

A Algoritam uvijek konvergira, međutim globalni maksimum log-izglednosti parametara doseže samo ako je broj grupa postavljen na pravi broj grupa ili tako da je broj grupa jednak broju primjera

B Algoritm uvijek konvergira, i to do točke u prostoru parametara koja maksimizira log-izglednost parametara, no brzina konvergencije ovisi o toma kako su inicijalizirani parametri

C Krenuvši od nekih početnih parametara, algoritam uvijek konvergira do parametara koji maksimiziraju očekivanje log-izglednosti, međutim to ne moraju biti parametri koji maksimiziraju vjerojatnost podataka

D Algoritam konvergira samo ako su primjeri u ulaznom prostoru sferični, ako su zavisnosti između značajki linearne, i ako nema multikolinearnosti, jer u protivnom zavisnosti nije moguće modelirati kovarijacijskom matricom

4. (T) Optimizaciju parametara modela Gaussove mješavine (GMM) ne provodimo u zatvorenoj formi. S druge strane, parametre Gaussovog Bayesovog klasifikatora, koji je sličan modelu GMM, optimiramo u zatvorenoj formi. **Zašto parametre GMM-a ne optimiramo u zatvorenoj formi, dok kod Gaussovog Bayesovog klasifikatora to radimo?**

A Za razliku od Gaussovog Bayesovog klasifikator, GMM je nenadizirani algoritam, pa log-izglednost podataka nije definirana i nije moguća maksimizacija u zatvorenoj formi

B Kod GMM-a, pored koeficijenata mješavine i vektora sredina, trebamo procijeniti i kovarijacijske matrice, za što ne postoji procjenitelj u zatvorenoj formi

C Kod GMM-a ne znamo koji primjer pripada kojoj grupi, pa je gustoća primjera jednaka zbroju gustoći komponenti, za što ne postoji maksimizator u zatvorenoj formi

D Parametri oba modela mogu se optimirati u zatvorenoj formi, međutim kod modela GMM računalno je jednostavnije koristiti EM-algoritam

5. (P) Algoritam GMM koristimo za grupiranje $N = 10$ primjera u dvodimenzijskome ulaznom prostoru. Skup primjera koje grupiramo je sljedeći:

$$\mathcal{D} = \{(0,0),(1,1),(1,2),(2,2),(2,3),(5,0),(5,1),(6,0),(6,6),(7,7)\}$$

Razmatramo tri modela GMM:

$$\begin{aligned}
\mathcal{H}_1 : &\quad K = 2 \text{ grupa, puna kovarijacijska matrica}\\
\mathcal{H}_2 : &\quad K = 2 \text{ grupa, izotropna kovarijacijska matrica}\\
\mathcal{H}_3 : &\quad K = 3 \text{ grupe, izotropna kovarijacijska matrica}
\end{aligned}$$

Za sva tri modela kovarijacijska matrica je nedijeljena, dakle svaka komponenta ima svoju kovarijacijsku matricu. Za početne centroide odabiremo nasumično dva odnosno tri primjera iz $\mathcal{D}$, ovisno o broju grupa K. Za svaki model grupiranje ponavljamo 100 puta te kao konačno grupiranje uzimamo ono s najvećom log-izglednošću na skupu $\mathcal{D}$. Zanima nas kojoj grupi najvjerojatnije pripada primjer $\mathbf{x}^{(5)} = (2,3)$, to jest zanima nas k koji maksimizira odgovornost $h_k^{(5)} = P(y=k|\mathbf{x}^{(5)})$. Ta vrijednost će biti različita za ova tri modela. Označimo sa h_α maksimalnu odgovornost za primjer $\mathbf{x}^{(5)}$ u modelu $\mathcal{H}_\alpha$, to jest vjerojatnost pripadanja tog primjera najvjerojatnijoj grupi dobivenoj grupiranjem pomoću modela $\mathcal{H}_\alpha$. **Što možemo zaključiti o odgovornostima h_α za ova tri modela?**

A $h_{\alpha_1} > h_{\alpha_2} > h_{\alpha_3}$ B $h_{\alpha_1} < h_{\alpha_2} < h_{\alpha_3}$ C $h_{\alpha_2} > h_{\alpha_1} > h_{\alpha_3}$ D $h_{\alpha_2} < h_{\alpha_1} < h_{\alpha_3}$

6. (P) Za grupiranje skupa primjera $\mathcal{D}$ koristimo algoritam GMM. Koristimo nekoliko varijanti tog modela:

$\mathcal{H}_1$: Model sa $K = 50$ središta inicijaliziranima algoritmom K-sredina

$\mathcal{H}_2$: Model sa $K = 50$ središta inicijaliziranima algoritmom K-sredina i dijeljenom kov. matricom

$\mathcal{H}_3$: Model sa $K = 50$ slučajno inicijaliziranim središtima i dijeljenom kov. matricom

$\mathcal{H}_4$: Model sa $K = 10$ središta inicijaliziranima algoritmom K-sredina i dijeljenom kov. matricom

Sa svakim modelom grupiranje ponavljamo 1000 puta i zatim za svaki model crtamo graf funkcije log-izglednosti kroz iteracije EM-algoritma, uprosječen kroz svih 1000 ponavljanja. Neka je LL_α^0 prosječna log-izglednost za model $\mathcal{H}_\alpha$ na početku izvođenja EM-algoritma, a neka je LL_α^* prosječna log-izglednost za taj model na kraju izvođenja EM-algoritma. **Što možemo unaprijed zaključiti o ovim log-izglednostima?**

A $LL_2^0 \geqslant LL_4^0$, $LL_1^* \geqslant LL_2^* \geqslant LL_3^*$

B $LL_3^0 \geqslant LL_4^0$, $LL_1^* \geqslant LL_3^* \geqslant LL_4^*$

C $LL_2^0 \geqslant LL_4^0 \geqslant LL_3^0$, $LL_1^* \geqslant LL_2^*$

D $LL_2^0 \leqslant LL_4^0$, $LL_2^* \leqslant LL_1^* \geqslant LL_3^*$

7. (T) Broj grupa K hiperparametar je mnogih algoritama grupiranja, pa tako i algoritma GMM. Optimalan broj grupa može se odrediti na razine načine, a jedan od njih je Akaikeov kriterij. **Na kojem se principu temelji odabir broja grupa Akaikeovim kriterijem?**

A Model s optimalnim brojem grupa je onaj koji podatke čini najvjerojatnijima, ali to čini sa što manje parametara

B Optimalan broj grupa je onaj kod kojeg, nakon daljnjeg povećanja broja grupa, vrijednost log-izglednosti stagnira ili blago raste

C Model s optimalnim brojem grupa je onaj koji minimizira log-izglednost nepotpunih podataka, a maksimizira log-izglednost potpunih podataka

D Optimalan broj grupa je onaj koji maksimizira očekivanje log-izglednost modela, uz pretpostavku izotropne kovarijacijske matrice

8. (N) Algoritmom hijerarhijskog aglomerativnog grupiranja (HAC) grupiramo $N = 5$ primjera. Za grupiranje koristimo mjeru sličnosti, koja je za naših pet primjera definirana sljedećom matricom (matrica je simetrična, pa je donji trokut izostavljen):

$$\begin{array}{c c}
 & \begin{array}{ccccc} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \mathbf{x}^{(3)} & \mathbf{x}^{(4)} & \mathbf{x}^{(5)} \end{array} \\
\begin{array}{c} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \\ \mathbf{x}^{(4)} \\ \mathbf{x}^{(5)} \end{array} &
\begin{pmatrix} 1 & 0.4 & 0.5 & 0.7 & 0.5 \\ & 1 & 0.9 & 0.3 & 0.6 \\ & & 1 & 0.7 & 0.1 \\ & & & 1 & 0.8 \\ & & & & 1 \end{pmatrix}
\end{array}$$

Provedite grupiranje algoritmom HAC s potpunim povezivanjem te nacrtajte pripadni dendrogram. Primijetite da dendrogram odgovara binarnom stablu, s pojedinim primjerima u listovima. **Kojem binarnom stablu odgovara dobiveni dendrogram?**

A $((\mathbf{x}^{(2)}, \mathbf{x}^{(3)}), \mathbf{x}^{(4)}), (\mathbf{x}^{(5)}, \mathbf{x}^{(1)}))$

B $((\mathbf{x}^{(2)}, \mathbf{x}^{(3)}), \mathbf{x}^{(1)}), (\mathbf{x}^{(4)}, \mathbf{x}^{(5)}))$

C $((\mathbf{x}^{(2)}, \mathbf{x}^{(3)}), ((\mathbf{x}^{(4)}, \mathbf{x}^{(5)}), \mathbf{x}^{(1)})))$

D $((\mathbf{x}^{(2)}, \mathbf{x}^{(3)}), ((\mathbf{x}^{(4)}, \mathbf{x}^{(1)}), \mathbf{x}^{(5)})))$