Model Gaussovih mješavina

Vrhovec, Veronika

Model Gaussovih mješavina

Vrhovec, Veronika

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Veronika Vrhovec

MODEL GAUSSOVIH MJEŠAVINA

Diplomski rad

Voditelj rada:
doc. dr. sc. Azra Tafro

Zagreb, 2020

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Ako želimo precizno opisati stvarni svijet koji nas okružuje, moramo ga sagledati iz više kuteva. Tako i kod modeliranja populacija koje opažamo često nije dovoljno koristiti jednodimenzionalnu analizu. A što ako nam ni višedimenzionalni pristup nije dovoljan da tu populaciju točno okarakteriziramo jer previdi postojanje podpopulacija? Intuitivno je tada modelirati svaku podpopulaciju zasebno, ali pritom voditi računa o njihovom međusobnom utjecaju. Teško je egzaktno tvrditi kada smo se prvi puta susreli s ovom vrstom problema u statističkoj analizi, ali već krajem 19. stoljeća nastaju prvi formalizirani modeli koji su primijenjeni na miješane populacije. Takve modele nazivamo modelima konačnih mješavina. Priroda nam je neiscrpan izvor ideja za znanstvena istraživanja pa ne čudi činjenica da je pozamašan broj modela konačnih mješavina primijenjen upravo na Gaussove mješavine, a one su predmet proučavanja ovog diplomskog rada. Mnoge prirodne i društvene znanosti koriste navedene modele mješavina u analizi populacija kako bi mogli usmjeriti istraživanje na komponente, umjesto da zanemare utjecaj podgrupa na distribucijske parametre. Gaussove mješavine naziv su dobile po Gaussovoj distribuciji budući da su njihove komponente normalno distribuirane i *najčešće* ih u modelima ima konačno mnogo. U ovom ćemo radu dati pregled statističkih metoda koje se koriste u njihovom modeliranju. Ovisno o populaciji koju proučavamo, broj komponenata može biti unaprijed poznat ili ga je potrebno procijeniti, a tada se koriste složenije numeričke metode. Dvije su glavne sastavnice modela konačnih mješavina. Prva je procjena parametara vjerojatnosne distribucije svih komponenata mješavine i određivanje pripadnih težina. Težine komponenata još nazivamo proporcijama miješanja jer nam daju uvid u udjele koji svaka komponenta ima u mješavini. Drugi ključan stavak modeliranja je utvrđivanje kojoj komponenti mješavine pripada svaka od jedinki promatrane populacije. Već iz navedenog vidimo da su ove analize međusobno zavisne i da ih moramo proučavati istovremeno. Problematici pristupamo analitički, a princip modeliranja prilagođavamo u ovisnosti od promatranog slučaja. U poglavlju 1 konstruiramo model konačnih mješavina pomoću teorije frekvencionističkog i Bayesovskog zaključivanja, pritom koristeći standardne metode procjene parametara. Teorija matematičke statistike korištena u konstrukciji modela uvedena je potpoglavljem 1.1. Konačne mješavine formalno su definirane u potpoglavlju 1.2 dok je u potpoglavlju 1.3 razrađen teorijski pristup statističkom zaključivanju u modelima Gausso-

vih mješavina. Vidjet ćemo da matematički alat izveden iz statističkog zaključivanja nije dovoljan za rješavanje najzanimljivijeg slučaja modeliranja, a to je slučaj kada zaključke o pripadnosti komponentama donosimo na temelju neopaženih podataka. Rješenje ovog problema nije se, kao u slučajevima s poznatim pridruživanjima, nametnulo samo od sebe. Bio je potreban dugi niz godina, ali i razvoj računarstva kao znanosti da bismo danas taj problem mogli jednostavno riješiti pomoću računalnog algoritma. Poglavlje 2 ovog rada rezervirano je za proučavanje računalnog pristupa modeliranju Gaussovih mješavina. Algoritam koji se koristi za pridruživanje jedinki komponentama je algoritam maksimizacije očekivanja te ga definiramo u potpoglavlju 2.1. Kako princip modeliranja mješavina ne bi ostao na teorijskoj razini, u potpoglavlju 2.2 ilustriramo navedene metode analizirajući stvarnu populaciju.

Poglavlje 1

Model konačnih mješavina

Model konačnih mješavina[1] je statistički model koji pretpostavlja postojanje neopaženih grupa unutar promatrane populacije. Problem modeliranja konačnih mješavina se svodi na analizu grupa i pridruživanje podataka grupama. Za svaku grupu treba odrediti svojstvene značajke te ispitati postoji li teorijska vjerojatnosna distribucija koja *dovoljno dobro* opisuje podatke iz te grupe. Prvi znanstveni radovi na ovu temu nastali su krajem 19. stoljeća te je fokus istraživanja bio modeliranje grupa, dok se pridruživanju podataka grupama nije davao veći značaj. Međutim, činjenica je da je grupiranje neodvojivo od potpune analize mješavine jer je utjecaj pridruživanja podataka na statističko zaključivanje[2] o tim podacima evidentan. U novije vrijeme, modeli konačnih mješavina prepoznati su u mnogim znanostima upravo zbog informativnosti o pridruživanju podataka grupama. Mogućnosti primjene modela su razne, ali o tome je opsežnije pisano u poglavlju 2. Ovo je poglavlje predviđeno za teorijsko objašnjenje matematičkog aparata koji koristimo pri modeliranju. Uvest ćemo teoriju iz vjerojatnosti i statistike koju ćemo nadograditi kako bismo mogli provesti metode modeliranja različitih populacija konačnih mješavina.

1.1 Vjerojatnosni prostor i slučajne veličine

Definirajmo objekte potrebne za uvođenje vjerojatnosnog prostora na $\mathbb{R}$ i navedimo teoriju matematičke statistike[3] koju ćemo koristiti u modeliranju u Poglavlju 2.

Definicija 1.1.1.

I. Neka je $\Omega \subseteq \mathbb{R}$ *neprazan* ***skup događaja****. Elementi skupa* Ω *su* ***elementarni događaji****.*

[1] eng. *finite mixture model*

[2] eng. *statistical inference*

[3] Definicije i teorijski rezultati preuzeti su iz Teorije vjerojatnosti [13], iz 2. i 9. poglavlja.

*II. Neka je $\mathcal{F} \subseteq \Omega$ familija skupova na Ω. $\mathcal{F}$ je **σ–algebra** ako vrijedi:*

(S1) $\emptyset \in \Omega$

(S2) $A \in \Omega \implies A^c \in \Omega$

(S3) $(A_n)_{n\in\mathbb{N}} \subseteq \Omega \implies \bigcup_{n=1}^{\infty} A_n \in \Omega$

*Uređeni par $(\Omega, \mathcal{F})$ nazivamo **izmjeriv prostor**.*

*III. Neka je $(\Omega, \mathcal{F})$ izmjerivi prostor i $\mathbb{P}\colon \mathcal{F} \to \mathbb{R}$ preslikavanje sa sljedećim svojstvima zovemo **vjerojatnost**:*

(P1) $\mathbb{P}(A) \geq 0$, *za svaki* $A \in \Omega$ ... *nenegativnost vjerojatnosti*

(P2) $\mathbb{P}(\Omega) = 1$ ... *normiranost vjerojatnosti*

(P3) $\mathbb{P}(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} \mathbb{P}(A_n)$ ... *σ–aditivnost vjerojatnosti*

*Uređenu trojku $(\Omega, \mathcal{F}, \mathbb{P})$ nazivamo **vjerojatnosni prostor**.*

Definicija 1.1.2. ***Borelova σ–algebra*** *na $\mathbb{R}$ je najmanja σ–algebra na $\mathbb{R}$ generirana familijom otvorenih skupova $\mathcal{U}$ na $\mathbb{R}$.*

$$\mathcal{B}_{\mathbb{R}} = \sigma(\mathcal{U}).$$

Definicija 1.1.3. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Preslikavanje $X\colon \Omega \to \mathbb{R}$ je **slučajna varijabla** ako vrijedi $X^{-1}(\mathcal{B}_{\mathbb{R}}) \subseteq \mathcal{F}$.*

Definicija 1.1.4. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i X slučajna varijabla na Ω. **Funkcija distribucije** slučajne varijable X je preslikavanje $F_X\colon \mathbb{R} \to [0,1]$ definirano s:*

$$F_X(x) = \mathbb{P}_X(\langle -\infty, x]) = \mathbb{P}\left(X^{-1}(\langle -\infty, x])\right) = \mathbb{P}(\omega \in \Omega : X(\omega) \leq x) = \mathbb{P}(X \leq x), \ \forall x \in \mathbb{R}$$

*Preslikavanje $\mathbb{P}_X$ nazivamo **zakon razdiobe** slučajne varijable X.*

Definicija 1.1.5. *Slučajna varijabla X na $(\Omega, \mathcal{F}, \mathbb{P})$ je **diskretna** ako postoji prebrojivi skup $D \subseteq \mathbb{R}$ takav da vrijedi $\mathbb{P}(X \in D) = 1$.*

Definicija 1.1.6. *Slučajna varijabla X na $(\Omega, \mathcal{F}, \mathbb{P})$ je **apsolutno neprekidna** ako za njezinu funkciju distribucije F_X postoji nenegativna Borelova funkcija $f_X\colon \mathbb{R} \to \mathbb{R}$ takva da vrijedi*

$$F_X(x) = \int_{\infty}^{x} f_X(t)\, d\lambda(t), \ x \in \mathbb{R}. \tag{1.1}$$

*Funkciju f_X tada nazivamo **funkcija gustoće** slučajne varijable X.*

Definicija 1.1.7. *Neka je X slučajna varijabla na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$*. Ako red* $\sum_{\omega\in\Omega} X(\omega)\mathbb{P}(\omega)$ *apsolutno konvergira, onda njegovu sumu zovemo* ***matematičko očekivanje slučajne varijable X*** *i označujemo s*

$$\mathbb{E}[X] = \sum_{\omega\in\Omega} X(\omega)\mathbb{P}(\omega). \tag{1.2}$$

Ako je X apsolutno neprekidna slučajna varijabla, onda je njeno očekivanje dano s

$$\mathbb{E}[X] = \int_{\Omega} X(\omega) f_X(\omega)\, \mathrm{d}\mathbb{P}(\omega). \tag{1.3}$$

Napomena 1.1.8. *Integral u definicijama 1.1.6 i 1.1.7 je Lebesgueov integral funkcije* f_X *u odnosu na Lebesgueovu mjeru* λ*.*

Definicija 1.1.9. *Slučajna varijabla X ima* ***normalnu distribuciju*** *s parametrima* $\mu \in \mathbb{R}$ *i* $\sigma \in \mathbb{R}^+$*,* $X \sim N(\mu, \sigma^2)$*, ako joj je funkcija gustoće dana s*

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),\ x \in \mathbb{R}. \tag{1.4}$$

Definicija 1.1.10 (Višedimenzionalna normalna distribucija)**.** *Neka je* $\boldsymbol{X} = (X_1, \ldots, X_n)$ *slučajni vektor, za* $n \in \mathbb{N}$*. Označimo s* $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in \mathbb{R}^n$ *očekivanje vektora* $\boldsymbol{X}$ *te sa* $\Sigma = \mathrm{Cov}(X_i, X_j) \in M_n(\mathbb{R})$ *kovarijacijsku matricu od* $\boldsymbol{X}$*. Stavimo da je* $\Sigma^{-1} = \left[\sigma_{ij}\right]$ *pa uvodimo kvadratnu formu* $Q_n \colon \mathbb{R}^n \to \mathbb{R}^+$ *sljedećim pravilom preslikavanja*

$$Q_n(\boldsymbol{x}) = Q_n(x_1, \ldots, x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij}(x_i - \mu_i)(x_j - \mu_j) = (\boldsymbol{x} - \boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})^{\tau}. \tag{1.5}$$

Kažemo da slučajni vektor $\boldsymbol{X}$ *ima* ***n-dimenzionalnu normalnu distribuciju****,* $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ *ako je funkcija gustoće od* $\boldsymbol{X}$ *dana s*

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}} (\det\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} Q_n(\boldsymbol{x})\right),\ \boldsymbol{x} \in \mathbb{R}^n. \tag{1.6}$$

Primijetimo da je ovako definirana kvadratna forma zapravo kvadratna Mahalanobisova udaljenost opažanja $\boldsymbol{x}$ od promatrane višedimenzionalne normalne distribucije, tj.

$$D_M(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})^{\tau}} = \sqrt{Q_n(\boldsymbol{x})}. \tag{1.7}$$

Kovarijacijska matrica Σ je po svojoj definiciji uvijek simetrična i pozitivno semidefinitna. U slučaju višedimenzionalne normalne distribucije slučajnog vektora $\boldsymbol{X}$, pripadna kovarijacijska matrica Σ je pozitivno definitna te je njezin inverz Σ^{-1} također simetrična i pozitivno definitna matrica.

Metode procjene distribucijskih parametara

Parametri promatrane vjerojatnosne distribucije su obično nepoznati te ih je potrebno odrediti statističkim metodama. Najčešće korištene metode procjene parametara su metoda momenata, metoda maksimalne vjerodostojnosti[4] i Bayesova procjena. Ovdje uvodimo princip korištenja tih metoda u općenitom slučaju, a u potpoglavlju 1.3 dajemo njihove izvode za procjenu parametara konačne mješavine. Neka je Y slučajna varijabla čija distribucija F_Y dolazi iz familije distribucija $\mathcal{T}(\boldsymbol{\theta})$ na skupu parametara $\boldsymbol{\Theta}$. Želimo procijeniti nepoznate parametre distribucije F_Y.

Metoda momenata

Neka je $\boldsymbol{y} = (y_1, \ldots, y_N)$ neko opažanje slučajne varijable Y, za $N \in \mathbb{N}$. Parametre distribucije F_Y procjenjujemo metodom momenata tako da izjednačimo uzoračke momente s teorijskim momentima razdiobe.

Definicija 1.1.11. ***Teorijski m–ti moment*** *je svojstvo funkcije distribucije slučajne varijable Y koje opisuje graf funkcije distribucije i definiramo ga s $\mathbb{E}[Y^m]$, gdje je $m \in \mathbb{N}$. Ako očekivanje $\mathbb{E}[Y^m]$ postoji, računamo ga formulom*

$$\mathbb{E}[Y^m] = \int_\Omega y^m dF_Y(y) = \int_\Omega y^m f_y(y) d\mathbb{P}. \tag{1.8}$$

Označimo s M_m **uzorački *m–ti moment*** koji odgovara m–tom teorijskom momentu $\mathbb{E}[Y^m]$. Pretpostavimo da je potrebno odrediti $k \in \mathbb{N}$ parametara promatrane distribucije. Tada rješavamo sustav od k jednadžbi oblika

$$M_m = \mathbb{E}[Y^m], \quad m = 1, \ldots, k. \tag{1.9}$$

Metoda maksimalne vjerodostojnosti

Uvedimo prvo definiciju funkcije vjerodostojnosti pa ćemo objasniti kako ovom metodom procijeniti nepoznate parametre distribucije.

Definicija 1.1.12. *Neka je $\boldsymbol{Y} = (Y_1, \ldots, Y_N)$, za $N \in \mathbb{N}$ slučajni uzorak iz modela $\mathcal{P} = \{f(\,\cdot\,; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ i $\boldsymbol{y} = (y_1, \ldots, y_N)$ neka njegova realizacija. Tada je* ***vjerodostojnost*** *definirana s*

$$L : \boldsymbol{\Theta} \to \mathbb{R}, \quad L(\boldsymbol{\theta} \mid \boldsymbol{y}) = L(\boldsymbol{\theta}) = f_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \prod_{k=1}^{N} f(y_k \mid \boldsymbol{\theta}) \tag{1.10}$$

[4] eng. *maximum likelihood estimation* — kraće **ML metoda**

Budući da je $f_{\boldsymbol{Y}}$ funkcija gustoće, ona je nenegativna funkcija pa možemo definirati ***log–vjerodostojnost*** *kao logaritam funkcije vjerodostojnosti, tj.*

$$l : \boldsymbol{\Theta} \to \mathbb{R}, \quad l(\boldsymbol{\theta} \mid \boldsymbol{y}) = l(\boldsymbol{\theta}) = \log f_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \sum_{k=1}^{N} \log f(y_k \mid \boldsymbol{\theta}) \tag{1.11}$$

Cilj procjene ML metodom je odrediti procjenitelj $\hat{\boldsymbol{\theta}}$ parametara $\boldsymbol{\theta}$ tako da određuje niz asimptotski efikasnih i konzistentnih korijena diferencijalne jednadžbe

$$\partial l(\boldsymbol{\theta}) \,/\, \partial \boldsymbol{\theta} = 0. \tag{1.12}$$

Limes vjerojatnosti korijena jednadžbe (1.12) teži u 1 pa ti korijeni odgovaraju lokalnom maksimumu u interioru prostora parametara.[5]. Definirajmo formalno procjenitelje spomenute u prethodnoj diskusiji. Iako procjenjujemo vektor parametara $\boldsymbol{\theta}$, zbog konteksta sljedećih definicija smatrat ćemo da procjenjujemo vrijednost nekog preslikavanja κ definiranog na prostoru parametara.

Definicija 1.1.13. *Procjenitelj $\hat{K} = K(\boldsymbol{Y})$ je* ***nepristrani procjenitelj*** *vrijednosti preslikavanja $\kappa(\boldsymbol{\theta})$ ako vrijedi*

$$\mathbb{E}\left[\hat{K}\right] = \kappa(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}. \tag{1.13}$$

Za definiciju efikasnog procjenitelja, potrebno je uvesti Fisherovu informaciju.

Definicija 1.1.14. ***Fisherova informacija*** *je preslikavanje definirano s*

$$I(\boldsymbol{\theta}) = \mathbb{E}\left[\left(\frac{\partial l\,(\boldsymbol{Y}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^2\right]. \tag{1.14}$$

Definicija 1.1.15. *Procjenitelj $\hat{K} = \kappa(\hat{\boldsymbol{\theta}})$ vrijednosti funkcije parametara $\kappa(\boldsymbol{\theta})$ je* ***efikasan*** *ako je nepristran za $\kappa(\boldsymbol{\theta})$, ima konačan drugi moment, te vrijedi*

$$Var(\hat{K}) = c\frac{(\kappa'(\boldsymbol{\theta}))^2}{I(\boldsymbol{\theta})}, \tag{1.15}$$

za neki $c > 0$. Procjenitelje koji uvjet efikasnosti postižu asimptotski nazivamo ***asimptotski efikasni procjenitelji****.*

Definicija 1.1.16. *Kažemo da je niz procjenitelja $(\hat{K}_n)_{n\in\mathbb{N}}$* ***konzistentan*** *za vrijednosti funkcije parametara $\kappa(\boldsymbol{\theta})$, ako konvergira[6] prema $\kappa(\boldsymbol{\theta})$.*

Može se pokazati da takav niz procjenitelja postoji kada su zadovoljeni Cramerovi uvjeti regularnosti koji su, dakle, nužan uvjet da smijemo koristiti ML metodu procjene.

[5]Detaljnije obrazloženje možemo pronaći u članku The EM Algorithm [9].

[6]Konvergencija može biti po vjerojatnosti, u srednjem reda $p \in \mathbb{N}$ ili gotovo svuda. (*kraće pišemo* ***g.s.***)

Uvjeti regularnosti:

- Parametar koji procjenjujemo je iz unutrašnjosti parametarskog prostora, $\boldsymbol{\theta} \in Int\ \boldsymbol{\Theta}$
- Log–vjerodostojnost je tri puta diferencijabilna na okolini stvarnog parametra $\boldsymbol{\theta}$ u parametarskom prostoru $\boldsymbol{\Theta}$, za g.s. $\boldsymbol{y} \in \Omega$
- Treća derivacija log–vjerodostojnosti je ograničena, $\left|\frac{\partial^3 l(\boldsymbol{\theta}|\boldsymbol{y})}{\partial \boldsymbol{\theta}^3}\right| < M, \quad$ za neki $M < \infty$, tj. Fisherova informacija je dobro definirana.

Definicija 1.1.17 (MLE procjenitelj[7])**.** *Statistika* $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\mathbf{Y})$ *je* ***procjenitelj maksimalne vjerodostojnosti*** *za parametar* $\boldsymbol{\theta}^\star \in \boldsymbol{\Theta}$ *ako vrijedi*

$$L(\hat{\boldsymbol{\theta}} \mid \mathbf{Y}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} L(\boldsymbol{\theta} \mid \boldsymbol{Y}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \prod_{k=1}^{N} f(y_k \mid \boldsymbol{\theta}). \tag{1.16}$$

Jedna od tehnika određivanja parametra $\boldsymbol{\theta}$ koji maksimizira funkciju vjerodostojnosti L je traženje njenih stacionarnih točaka i klasifikacija na sedlastu točku, globalni ili lokalni ekstrem. Funkcija vjerodostojnosti je produkt funkcija gustoća modela $\mathcal{P}$ pa, zbog diferenciranja, račun određivanja točaka ekstrema funkcije postaje zahtjevan. Ovisno o promatranoj funkciji gustoće, moguć je slučaj da se problem ne može riješiti analitički. Funkcija logaritmiranja je neopadajuća na svojoj domeni, stoga funkcije (1.10) i (1.11) postižu maksimum za isti $\boldsymbol{\theta}$, tj.

$$l(\hat{\boldsymbol{\theta}} \mid \mathbf{Y}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} l(\boldsymbol{\theta} \mid \mathbf{Y}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{k=1}^{N} \log(f(y_k \mid \boldsymbol{\theta})). \tag{1.17}$$

Zbog jednostavnijeg računa, u nastavku rada ML procjenu parametara distribucije provodimo tako da odredimo MLE procjenitelj funkcije log–vjerodostojnosti. Račun je osobito jednostavan u slučaju da promatrana distribucija pripada eksponencijalnoj familiji distribucija. U procjeni parametara Bayesovom metodom, potrebno nam je statističko zaključivanje bazirano na opaženim podacima iz modela, a izvedeno je iz sljedećeg teorema.

Teorem 1.1.18. ***(Bayesov teorem***[8]*)** *Neka je* $\boldsymbol{\theta}$ *apriorno nepoznati skup parametara promatrane distribucije i* Y *skup mogućih realizacija opažanja. Funkciju gustoće distribucije uvjetno na opažanje računamo formulom*

$$f(\boldsymbol{\theta} \mid Y) = \frac{f(Y, \boldsymbol{\theta})}{f(Y)} = \frac{f(Y \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(Y)}. \tag{1.18}$$

[7] eng. *maximum likelihood estimator*

[8] Formulacija Bayesovog teorema preuzeta iz Bayesian Statistics and Marketing [12], iz 2. poglavlja.

U formuli (1.18) uveli smo ***apriornu gustoću distribucije*** $f(Y)$ i ***apriornu vjerodostojnost*** $f(Y \mid \boldsymbol{\theta})$. Funkciju $f(\boldsymbol{\theta} \mid Y)$ zovemo ***aposteriorna gustoća distribucije***. Bayesovsko zaključivanje koristi se u slučajevima kada želimo izvesti zaključak o mogućim vrijednostima parametra $\boldsymbol{\theta}$ na temelju podataka opaženih u uzorku. Primijetimo da je nazivnik u jednadžbi (1.18) konstantan jer ne ovisi o $\boldsymbol{\theta}$. Aposteriornu gustoću stoga analiziramo zanemarujući nazivnik u jednadžbi (1.18), tj. prema relaciji proporcionalnosti

$$f(\boldsymbol{\theta} \mid Y) \propto f(Y \mid \boldsymbol{\theta}) f(\boldsymbol{\theta}). \tag{1.19}$$

Konstanta proporcionalnosti je marginalna gustoća uzorka koju računamo formulom

$$f(Y) = \int_{\boldsymbol{\Theta}} f(Y, \hat{\boldsymbol{\theta}}) d\hat{\boldsymbol{\theta}} = \int_{\boldsymbol{\Theta}} f(Y \mid \hat{\boldsymbol{\theta}}) f(\hat{\boldsymbol{\theta}}) d\hat{\boldsymbol{\theta}}. \tag{1.20}$$

Bayesova metoda

Neka je $f(\boldsymbol{\theta})$ apriorna distribucija nepoznatog parametra $\boldsymbol{\theta}$. Označimo s $\hat{\boldsymbol{\theta}}$ procjenitelj od $\boldsymbol{\theta}$ te uvedimo ***funkciju gubitka***[9] $\lambda = \lambda(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$. Funkcija gubitka je svojevrsna mjera udaljenosti procijenjenog parametra od njegove stvarne vrijednosti. Često se koriste mjere apsolutnog i srednjekvadratnog odstupanja koje računamo formulama:

$$\lambda(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \begin{cases} \left| \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right|, & \text{apsolutno odstupanje} \\ \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right)^2, & \text{srednjekvadratno odstupanje} \end{cases} \tag{1.21}$$

Bayesova funkcija rizika definirana je kao očekivanje funkcije gubitka, $R(\boldsymbol{\theta}) = \mathbb{E}\left[\lambda(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\right]$.

Definicija 1.1.19. *Procjenitelj $\hat{\boldsymbol{\theta}}$ je **Bayesov procjenitelj** ako minimizira Bayesovu funkciju rizika na prostoru parametara $\boldsymbol{\Theta}$.*

Prije no što nekom od metoda odredimo parametre distribucije, trebalo bi razmisliti je li uzorak na osnovu čijih opažanja donosimo zaključke dovoljno dobar. Je li moguće iz prirodnih pretpostavki promatranog modela izvesti dobre i *točne* zaključke? Odgovore na ta pitanja daje nam ***raspoznatljivost modela***. Raspoznatljivost modela je svojstvo da iz beskonačno mnogo provedenih opažanja možemo jednoznačno odrediti stvarne teorijske vrijednosti inherentnih parametara modela. Svojstvo jednoznačnog određivanja parametara ekvivalentno je svojstvu da različite vrijednosti parametara modela generiraju različite vjerojatnosne distribucije opaženih podataka. To znači da je na raspoznatljivim mješavinama moguće provesti točnu procjenu parametara i shodno tome statističko zaključivanje na temelju opaženih podataka.

[9] eng. *loss function*; u literaturi se najčešće označava s L, ali u ovom je radu ta oznaka već rezervirana za funkciju vjerodostojnosti

Definicija 1.1.20. *Neka je $\mathcal{P}$ parametarska familija distribucija indeksirana parametrom $\boldsymbol{\vartheta} \in \boldsymbol{\Theta}$ koji je definiran na prostoru $\mathcal{Y}$. Familija $\mathcal{P}$ je* ***raspoznatljiva***[10] *ako vrijedi*

$(\forall \boldsymbol{\vartheta}, \boldsymbol{\vartheta}^* \in \boldsymbol{\Theta})$ *parametri $\boldsymbol{\vartheta}$ i $\boldsymbol{\vartheta}^*$ definiraju istu vjerojatnost na $\mathcal{Y}$ ako i samo ako je $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}^*$.*

Raspoznatljivost familije distribucija možemo primijeniti i na funkcije gustoće pa vrijedi

$$f(\mathbf{y} \mid \boldsymbol{\vartheta}) = f(\mathbf{y} \mid \boldsymbol{\vartheta}^*), \text{ za g.s. } \mathbf{y} \in \mathcal{Y} \quad \Longrightarrow \quad \boldsymbol{\vartheta} = \boldsymbol{\vartheta}^*. \tag{1.22}$$

Skup $\mathcal{U}(\boldsymbol{\vartheta}) = \{\boldsymbol{\vartheta}^* \in \boldsymbol{\Theta} \colon f(\mathbf{y} \mid \boldsymbol{\vartheta}^*) = f(\mathbf{y} \mid \vartheta), \text{ za g.s. } \mathbf{y} \in \mathcal{Y}\} \subseteq \boldsymbol{\Theta}$ je ***neraspoznatljiv skup*** ako nije jednočlan, a pripadna familija distribucija $\mathcal{T}_{\mathcal{U}}(\boldsymbol{\vartheta})$ je tada ***neraspoznatljiva***.

1.2 Konačne mješavine

Kako bismo konstruirali model koji najbolje opisuje populaciju koju proučavamo, trebaju nam početne pretpostavke o prirodi tih podataka, o mogućoj pripadnosti nekoj vjerojatnosnoj distribuciji ili o postojanju smislene podjele na podpopulacije koje još nazivamo grupe ili komponente. Komponente bi trebale dijeliti populaciju na način da podaci budu homogeni unutar i heterogeni izvan njih. Za proizvoljni slučajni uzorak opažamo neku realizaciju i želimo znati kojoj komponenti populacije ona pripada, ali tu informaciju ne dobivamo izravno. Problem je riješen uvođenjem modela mješavina koje još nazivamo modelima miješanih gustoća[11]. Prvi rad u kojem je spomenuto pridruživanje opažanja nekoj od grupa populacije napisao je W. Feller 1943. godine u članku *On a General Class of "Contagious" Distributions* [3] u kontekstu modeliranja zaraze populacije ličinki. U ovom radu modeliramo konačne Gaussove mješavine[12], tj. pretpostavljamo da slučajni uzorak pripada populaciji s konačno mnogo normalno distribuiranih podpopulacija. Postoje modeli miješanih gustoća koji imaju beskonačno mnogo komponenata, a komponente općenito mogu pripadati bilo kojoj diskretnoj ili neprekidnoj razdiobi. Svi izvodi i zaključivanja se lako prilagođavaju diskretnom slučaju korištenjem brojeće mjere. Glavni cilj analiziranja konačnih mješavina je određivanje parametara distribucije svake komponente kako bi se daljnje statističko zaključivanje moglo prilagoditi predmetnoj populaciji.

Definicija 1.2.1. *Neka je $\boldsymbol{Y} = (Y_1, \ldots, Y_r)$ r–dimenzionalni slučajni vektor definiran u vjerojatnosnom prostoru $(\mathcal{Y}, \mathcal{F}, \mathbb{P})$, gdje je $r \in \mathbb{N}$. Promatramo $N \in \mathbb{N}$ nezavisnih opažanja slučajnog vektora $\boldsymbol{Y}$ koja se sastoje od r značajki. Označimo ih s $\mathbf{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$, gdje je $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ir})$. Kažemo da $\boldsymbol{Y}$ dolazi iz* ***distribucije konačne mješavine*** *ako njena vjerojatnosna funkcija gustoće $f_{\boldsymbol{Y}}$ ima formu gustoće mješavine*

[10] eng. *identifiable*; definicija preuzeta iz Finite Mixture and Markov Switching Models [4], str. 14.–15.

[11] eng. *mixture models*

[12] Definicije i izvodi modela konačnih mješavina preuzeti su iz Finite Mixture and Markov Switching Models [4], iz 2. poglavlja.

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \eta_1 f_1(\boldsymbol{y}) + \cdots + \eta_K f_K(\boldsymbol{y}), \quad \textit{za svaki } \boldsymbol{y} \in \boldsymbol{Y}, \tag{1.23}$$

gdje je $K \in \mathbb{N}$ broj komponenata, a f_k su komponentne gustoće. Vektor $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_K)$ je distribucija težina, gdje je težina komponente $\mathcal{K}_k$ označena parametrom η_k. Komponente vektora $\boldsymbol{\eta}$ su nenegativne i normirane, tj. takve da vrijedi

$$\eta_k \geq 0, \quad \forall k \in \{1, \ldots, K\} \tag{1.24}$$

$$\eta_1 + \cdots + \eta_K = 1. \tag{1.25}$$

Parametar težine k–te komponente možemo, stoga, interpretirati kao vjerojatnost da slučajno odabrana jedinka Y iz te populacije pripada komponenti $\mathcal{K}_k$. Stavimo $\mathbb{P}(Y \in \mathcal{K}_k) = \eta_k$. Zbog svojstva (1.25) normiranosti vektora težina, ne procjenjujemo svih K težina komponenata već je dovoljno procijeniti njih $K - 1$.

Neka je slučajni vektor $\boldsymbol{Y} = (Y_1, \ldots, Y_r)$ distribuiran iz mješavine K višedimenzionalnih normalnih distribucija, tj. $Y_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$, za $k \in \{1, \ldots, K\}$. Tada je $\boldsymbol{\mu}_k \in \mathbb{R}^r$, $\Sigma_k \in M_r(\mathbb{R})$ te je komponentna funkcija gustoće

$$f_k(\boldsymbol{y}) = (2\pi)^{-\frac{r}{2}} (\det \Sigma_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu_k})\Sigma_k^{-1}(\boldsymbol{y} - \boldsymbol{\mu_k})^\tau\right), \ \boldsymbol{y} \in \mathbb{R}^r. \tag{1.26}$$

Parametri Gaussove mješavine koje procjenjujemo su $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \eta_1, \ldots, \eta_K)$, gdje je $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \Sigma_k)$. Vektor očekivanja od $\boldsymbol{Y}$ je $\boldsymbol{\mu} = (\boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_K})$, gdje je $\boldsymbol{\mu_k} = (\mu_{k1}, \ldots, \mu_{kr})$ očekivanje komponente $\mathcal{K}_k$, za $k \in \{1, \ldots, K\}$. U definiciji 1.1.10 smo uveli oznaku Σ_k za kovarijacijsku matricu k–te komponente mješavine pa vektor kovarijacijskih matrica komponenata označimo sa $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K)$.

Pri modeliranju konačnih mješavina, osim procjene parametara modela, želimo odrediti pripadnost opažanja pojedinim komponentama. Pritom se susrećemo s opaženim i skrivenim varijablama.[13]

Definicija 1.2.2. ***Manifestna*** *ili opažena* ***varijabla*** *je slučajna varijabla čiju realizaciju opažamo iz uzorka i koja upućuje na prisustvo neke latentne*[14] *varijable.*

Definicija 1.2.3. ***Latentna*** *ili skrivena* ***varijabla*** *je slučajna varijabla čiju realizaciju ne opažamo izravno. Njeno postojanje se pretpostavlja s ciljem objašnjavanja manifestnih varijabli. Model koji objašnjava opažene varijable u terminima latentnih, naziva se* ***model latentnih varijabli****.*

[13] Definicija preuzeta iz materijala kolegija Strojno učenje [18]

[14] lat. *lateo* – biti skriven

Komponente modela konačnih mješavina $\mathcal{K}_1, \ldots, \mathcal{K}_K$ indeksiramo konačnim diskretnim skupom indikatora $\mathcal{I} = \{1, \ldots, K\}$, gdje je $K \in \mathbb{N}$. Pretpostavimo da za slučajni vektor $\boldsymbol{Y_i}$ iz promatrane populacije apriorno znamo da pripada komponenti $\mathcal{K}_k$. Njegova je funkcija gustoće $f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k)$ tada dana uvjetno na grupu k, gdje je $\boldsymbol{\theta}_k$ vektor parametara distribucije komponente $\mathcal{K}_k$. Zajedničku gustoću sada možemo izraziti formulom

$$f(\boldsymbol{y_i}, s_i = k) = f(\boldsymbol{y_i} \mid s_i = k)\mathbb{P}(s_i = k) = f(\boldsymbol{y_i} \mid s_i = k)\eta_k. \tag{1.27}$$

Uveli smo pojam modela latentnih varijabli, a model konačnih mješavina ćemo promatrati kao model hijerarhijske latentne varijable. Distribucija opažanja $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ ovisi o realizaciji $\boldsymbol{s} = (s_1, \ldots, s_N)$ latentnog diskretnog vektora $\boldsymbol{S} = (S_1, \ldots, S_N)$. Slučajni vektor $\boldsymbol{S}$ nazivamo ***pridruživanje*** jer sadrži informaciju o pridruživanju svakog od N opažanja odgovarajućoj komponenti. Skup $\{\boldsymbol{y}, \boldsymbol{s}\}$ zovemo ***potpun skup podataka***, dok je $\{\boldsymbol{y}\}$ ***nepotpun skup podataka***. Kod modeliranja konačnih mješavina, autori članka *Finite Mixture Models* [7] zaključuju da vektor pridruživanja $\boldsymbol{S}$ možemo gledati kao N latentnih jednako distribuiranih slučajnih varijabli. Potrebna je pretpostavka da su pridruživanja svakog od opažanja međusobno nezavisna te se može pokazati da je njihova razdioba multinomijalna,

$$S_k \sim Mult_K(1, \boldsymbol{\eta}), \quad \forall k \in \{1, \ldots, N\}.$$

Marginalna gustoća slučajne veličine $\boldsymbol{Y}$ je, u slučaju da apriorno nemamo pridruživanje $\boldsymbol{S}$, dana sljedećom linearnom kombinacijom funkcija gustoća:

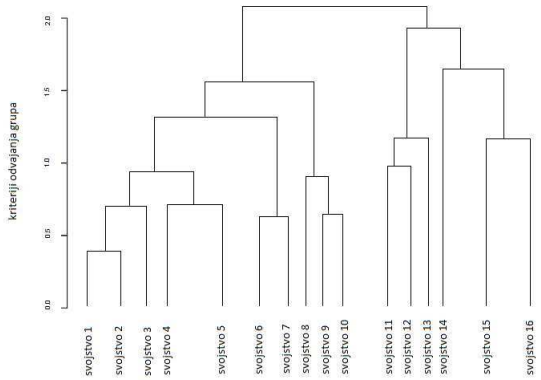
$$f(\boldsymbol{y}) = \sum_{k=1}^{K} f(\boldsymbol{y}, s = k) = \eta_1 f(\boldsymbol{y} \mid \boldsymbol{\theta}_1) + \cdots + \eta_K f(\boldsymbol{y} \mid \boldsymbol{\theta}_K), \tag{1.28}$$

gdje je s oznaka komponente kojoj opažanje $\boldsymbol{y}$ pripada. U slučaju Gaussove mješavine, komponentna funkcija gustoće $f(\boldsymbol{y} \mid \boldsymbol{\mu}_k, \Sigma_k)$ je gustoća višedimenzionalne normalne distribucije $\mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$ uvedene jednadžbom (1.26).

Hijerarhijski modeli

U ovom radu nepoznate parametre miješane gustoće procjenjujemo pomoću Bayesovskog zaključivanja i maksimizacijom log–vjerodostojnosti. Model Gaussove mješavine pripada klasi hijerarhijskih modela. To su modeli grupiranja u kojima sortiramo prema hijerarhijskoj relaciji po slojevima, tako da je za svaki par slojeva poznat uređaj među njima. Grafički možemo odnose između grupa u modelu prikazati dendrogramom. Hijerarhijske modele prema načinu grupiranja dijelimo na aglomerativne i divizivne. Aglomerativni modeli u prvom koraku grupiraju svako opažanje u zasebnu komponentu. Komponente se, kroz iteracije algoritma grupiranja, postepeno stapaju u zajedničke grupe te je na kraju njihov broj manji nego na početku. Divizivni modeli imaju obrnutu logiku i kreću od jedne zajedničke grupe koju dijele na komponente. Broj komponenata divizivnog modela je po

završetku grupiranja veći nego što je bio na početku. U poglavlju 2 objasnit ćemo detaljno algoritam maksimizacije očekivanja[15] koji je primjer aglomerativnog hijerarhijskog modela i koji koristimo za grupiranje konačnih mješavina.



Slika 1.1: dendrogram

Aglomerativni model prikazan ovim dendrogramom možemo interpretirati tako da postoje tri osnovne podgrupe populacije te da je svojstvo 14 netipična vrijednost.

Definicija 1.2.4. ***Bayesov hijerarhijski model*** *je statistički model procjene parametara aposteriorne vjerojatnosne distribucije korištenjem Bayesovskog zaključivanja na apriornim vjerojatnosnim distribucijama prvog i drugog sloja modela.*

U prvom sloju dana je zajednička apriorna distribucija opažanja uvjetno na apriorno poznatu realizaciju pridruživanja $\boldsymbol{s}$:

$$f(\mathbf{y} \mid \mathbf{s}, \boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(\mathbf{y}_i \mid s_i, \boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(\mathbf{y}_i \mid \boldsymbol{\theta}_{S_i}). \tag{1.29}$$

Drugi sloj modela daje funkciju gustoće zajedničke distribucije $f(\mathbf{s} \mid \boldsymbol{\vartheta})$ svih, u tom trenu nepoznatih realizacija $\boldsymbol{s}$ i to je diskretna distribucija duž rešetke $\mathcal{S}_K$.

$$\mathcal{S}_K = \{(S_1, \ldots, S_N) \colon S_i \in \{1, \ldots, K\},\ i = 1, \ldots, N\} \tag{1.30}$$

Uz pretpostavku da su indikatori $s_1, \ldots, s_N$ međusobno nezavisni, imamo da je vjerojatnost $\mathbb{P}(s_i = k \mid \boldsymbol{\eta}) = \eta_k$ pa je zajednička funkcija distribucije uvjetno na poznate težine

[15] eng. *expectation maximization algorithm*, kraće **EM algoritam**

komponenata dana izrazom

$$f(\mathbf{s} \mid \boldsymbol{\eta}) = \prod_{i=1}^{N} f(s_i \mid \boldsymbol{\eta}). \tag{1.31}$$

Konačno, koristimo Bayesovo pravilo (1.18) kao rezultat Bayesovog teorema 1.1.18 da izvedemo aposteriornu gustoću zajedničke distribucije $f(\boldsymbol{s} \mid \boldsymbol{\vartheta}, \boldsymbol{y})$ pomoću vjerodostojnosti $f(y \mid \boldsymbol{s}, \boldsymbol{\vartheta})$ i apriorne gustoće zajedničke distribucije $f(\boldsymbol{s} \mid \boldsymbol{\vartheta})$:

$$f(\boldsymbol{s} \mid \boldsymbol{\vartheta}, \boldsymbol{y}) \propto f(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{\vartheta}) f(\boldsymbol{s} \mid \boldsymbol{\vartheta}). \tag{1.32}$$

1.3 Statističko zaključivanje u modelu Gaussovih mješavina

U ovom potpoglavlju izvodimo statističko zaključivanje za model konačnih mješavina, na način da će, uz račun za općeniti model, biti izveden i rezultat za konačnu Gaussovu mješavinu. Pretpostavka za oba modela je da se mješavine sastoje od $K \in \mathbb{N}$ distribucija iz predmetne familije distribucije. Neka je $\mathcal{T}(\boldsymbol{\theta})$ familija distribucija na skupu parametara $\boldsymbol{\Theta}$ i neka je $\mathcal{G}(\boldsymbol{\theta}) \subseteq \mathcal{T}(\boldsymbol{\theta})$ familija normalnih distribucija. Označimo s $\boldsymbol{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$, $N \in \mathbb{N}$ promatranu realizaciju slučajnog uzorka iz konačne mješavine. Apriorne funkcije gustoće $f(\boldsymbol{y} \mid \boldsymbol{\theta})$ indeksirane su parametrom $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ te ih računamo po formuli

$$f(\boldsymbol{y_i} \mid \boldsymbol{\vartheta}) = \sum_{k=1}^{K} \eta_k f_k(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k). \tag{1.33}$$

U slučaju modeliranja višedimenzionalne Gaussove mješavine, komponentna funkcija gustoće ima formulu (1.26), a njezin parametar koji procjenjujemo je $\boldsymbol{\theta_k} = (\boldsymbol{\mu}_k, \Sigma_k)$. Problemu procjene parametara distribucije za svaku od komponenata pristupamo u tri koraka, kao što je navedeno u potpoglavlju 1.2:

1. Određivanje broja komponenata mješavine, $K \in \mathbb{N}$.
2. Procjena parametara $\boldsymbol{\theta}$ komponenata mješavine i vektora težine $\boldsymbol{\eta}$.
3. Pridruživanje svakog opažanja iz uzorka nekoj komponenti te zaključivanje na temelju latentnog indikatora $\boldsymbol{S}$.

Implementirajući model konačnih mješavina na stvarni poslovni ili znanstveni problem, najčešće na početku studije nemamo informaciju o broju komponenata populacije

koju modeliramo te ga moramo procijeniti. Neke od često korištenih metoda su testiranje prikladnosti modela[16] dobivenog metodom momenata te Bayesovsko zaključivanje uz pretpostavku neodređenosti modela. Nakon što je broj komponenata populacije određen, krećemo s procjenom parametara mješavine i pridruživanjem opažanja. Po završetku modeliranja, potrebno je izračunati utjecaj pretpostavljenog broja komponenata K na pogrešku u zaključivanju. Pritom se za ocjenu pogreške najčešće koriste Akaikeov, odnosno Bayesov informacijski kriterij (procjenitelj pogreške označavamo AIC, odn. BIC). U nastavku rada pretpostavljamo da je broj komponenata promatrane populacije unaprijed poznat i označimo ga s $K \in \mathbb{N}$. Za procjenu distribucijskih parametara komponenata mješavine koristimo EM algoritam. Opažanja pridružujemo odgovarajućim komponentama mješavine, uz dani procijenjeni vektor $\boldsymbol{\vartheta}$, tako da izračunamo aposteriornu vjerojatnost zajedničke funkcije distribucije $f(\boldsymbol{s} \mid \boldsymbol{\vartheta})$. Svakom od $N \in \mathbb{N}$ opažanja dodjeljujemo oznaku one komponente čija je vjerojatnost $f(s_i \mid \vartheta_j)$ najveća, za sve indekse $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, K\}$. Statističko zaključivanje moramo prilagoditi dostupnim informacijama o populaciji koju modeliramo, stoga je ovo potpoglavlje podijeljeno u tri dijela.[17] Prvo analiziramo najjednostavniji slučaj, kada su poznati parametri $\boldsymbol{\vartheta}$ te je potrebno pridružiti opažanja komponentama. Zatim obrađujemo procjenu parametara u slučaju kada je poznato pridruživanje vektora opažanja. Zadnji je generalizirani slučaj kada ne postoje apriorne pretpostavke pa je tako i pridruživanje opažanja apriorno nepoznato.

Pridruživanje opažanja u slučaju poznatih komponenata

Pretpostavimo da je poznat $\boldsymbol{\vartheta}$, tj. da su poznati vektor težina $\boldsymbol{\eta}$ i parametri distribucije $\boldsymbol{\theta}$ konačne mješavine koja se sastoji od K komponenata. Tada je apriorna funkcija gustoće $f(\boldsymbol{y} \mid \boldsymbol{\theta})$ egzaktno određena formulom (1.33). Neka je $\mathbf{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$ vektor opažanja slučajnog uzorka iz promatrane mješavine. Budući da su poznati parametri komponenata modela, preostalo je još samo odrediti pridruživanje vektora $\boldsymbol{y}$ komponentama. Problem pridruživanja opažanja riješen je korištenjem Bayesovog pravila (1.18). Uveli smo vektor pridruživanja $\boldsymbol{s} = (s_1, \ldots, s_N)$ kao realizaciju diskretnog slučajnog vektora indikatora s vrijednostima u skupu $\{1, \ldots, K\}$. Realizacija slučajne varijable S_i označava kojoj komponenti pripada opažanje $\boldsymbol{y_i}$, tj.

$$s_i = k \cdot \mathbb{1}_{\{\boldsymbol{y_i} \in \mathcal{K}_k\}}. \tag{1.34}$$

Pridruživanje vektora opažanja $\boldsymbol{y}$ ekvivalentno je pridruživanju svakog pojedinog opažanja u slučaju kada su poznati svi parametri mješavine. Problem je složeniji u slučaju da su parametri neke komponente nepoznati, ali je rješiv korištenjem Bayesovog klasifikatora. U

[16] eng. *goodness of fit test*

[17] Svi izvodi potpoglavlja 1.3 prate konstrukciju zaključivanja iz knjige Finite Mixture and Markov Switching Models [4] iz 2. poglavlja.

slučaju pridruživanja jednog opažanja $\boldsymbol{y_i}$, odmah krećemo s računanjem aposteriorne vjerojatnosti događaja $\{s_i = k\}, k \in \mathcal{I}$ uvjetno na opažanje $\{\boldsymbol{Y} = \boldsymbol{y_i}\}$ i na poznavanje distribucije mješavine. Vjerojatnost računamo formulom

$$\mathbb{P}(s_i = k \mid \boldsymbol{y_i}, \boldsymbol{\vartheta}) = \frac{\mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = k, \boldsymbol{\vartheta})\mathbb{P}(s_i = k \mid \boldsymbol{\vartheta})}{\sum_{j=1}^{K} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = j, \boldsymbol{\vartheta})\mathbb{P}(s_i = j \mid \boldsymbol{\vartheta})}. \tag{1.35}$$

Pritom je poznata apriorna vjerojatnost da opažanje $\boldsymbol{y_i}$ pripada komponenti $\mathcal{K}_k$ i ona je prema definiciji 1.2.1 jednaka $\mathbb{P}(s_i = k \mid \boldsymbol{\vartheta}) = \eta_k$. Uvrstimo η_k u jednadžbu (1.35):

$$\mathbb{P}(s_i = k \mid \boldsymbol{y_i}, \boldsymbol{\vartheta}) = \frac{\mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = k, \boldsymbol{\vartheta})\eta_k}{\sum_{j=1}^{K} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = j, \boldsymbol{\vartheta})\eta_j} = h_{ik}. \tag{1.36}$$

Primijetimo da izraz u nazivniku ne ovisi o konkretnom k. Za izvedenu aposteriornu vjerojatnost pridruživanja i–tog opažanja k–toj komponenti je uvedena oznaka h_{ik}. Ova vjerojatnost naziva se još i ***odgovornost***[18]. Budući da opažanje $\boldsymbol{y_i}$ alociramo onoj komponenti $\mathcal{K}_k$ čija je vjerojatnost (1.36) najveća, dovoljno je, kao u izvodu Bayesovog teorema 1.1.18, promatrati relaciju

$$\mathbb{P}(s_i = k \mid \boldsymbol{y_i}, \boldsymbol{\vartheta}) \propto \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = k, \boldsymbol{\vartheta})\eta_k. \tag{1.37}$$

Klasifikacijski model koji ovdje koristimo je ***Bayesov naivni klasifikator***[19]. Radi se o parametarskom modelu pridruživanja opažanja komponentama koji koristi Bayesovsko zaključivanje, a termin *naivni* dolazi od pretpostavke da su sva opažanja međusobno nezavisna uvjetno na promatranu komponentu $\mathcal{K}$, tj. vrijedi

$$\mathbb{P}(\boldsymbol{y_i} \mid \boldsymbol{y_j}, \mathcal{K}) = \mathbb{P}(\boldsymbol{y_i} \mid \mathcal{K}), \quad \forall i, j \in \{1, \ldots, N\},\ i \neq j. \tag{1.38}$$

Izvedimo sad postupak pridruživanja čitavog vektora opažanja $\boldsymbol{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$ koji ćemo zvati ***zajedničko pridruživanje***. Zanima nas vjerojatnost događaja $\{S_1 = s_1, \ldots, S_N = s_N\}$, za sva moguća pridruživanja $(s_1, \ldots, s_N)$ svakog od N opažanja nekoj od K komponenata. Relacija vjerojatnosti pridruživanja jednog opažanja (1.37) poopćena na čitav uzorak je

$$f(\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{\vartheta}) \propto f(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{\vartheta}) f(\boldsymbol{s} \mid \boldsymbol{\vartheta}), \tag{1.39}$$

gdje je $f(\boldsymbol{s} \mid \boldsymbol{\vartheta})$ apriorna gustoća zajedničke distribucije vektora indikatora $\boldsymbol{S} = (S_1, \ldots, S_N)$ *trenutno* nepoznatih pridruživanja i ona je poznata prije određivanja vektora opažanja $\boldsymbol{y}$. Vjerojatnost $f(\boldsymbol{y} | \boldsymbol{s}, \boldsymbol{\vartheta})$ je vjerodostojnost uvedena u relaciji (1.32) i ona je gustoća uzoračke distribucije čitavog vektora opažanja $\boldsymbol{y}$ pod uvjetom poznatog pridruživanja $\boldsymbol{s}$ i poznatih komponenata mješavine $\boldsymbol{\vartheta}$. Koristimo pretpostavku nezavisnosti slučajne varijable $\boldsymbol{Y}$ koja pripada promatranoj mješavini pa onda vjerodostojnost računamo formulom

[18] eng. *responsibility*

[19] eng. *naïve Bayes' classifier*; Teorija je izvedena iz materijala kolegija Strojno učenje [18].

$$f(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{\vartheta}) = f(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) = \prod_{i=1}^{N} f(y_i \mid \boldsymbol{\theta}_{S_i}). \tag{1.40}$$

Uz pretpostavku da su pridruživanja opažanja također međusobno nezavisna, apriorna gustoća zajedničke distribucije je onda

$$f(\boldsymbol{s} \mid \boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(s_i \mid \boldsymbol{\vartheta}). \tag{1.41}$$

Konačno imamo sve potrebno za primjenu Bayesovog pravila (1.18) i izračun aposteriorne gustoće zajedničke distribucije $f(\boldsymbol{s} \mid \boldsymbol{\vartheta}, \boldsymbol{y})$

$$f(\boldsymbol{s} \mid \boldsymbol{\vartheta}, \boldsymbol{y}) = \prod_{i=1}^{N} f(s_i \mid \boldsymbol{\vartheta}, \boldsymbol{y_i}). \tag{1.42}$$

Prema diskusiji s početka ove cjeline, aposteriornu gustoću zajedničke distribucije vektora indikatora $\boldsymbol{S}$ uvjetno na poznate parametre mješavine $\boldsymbol{\vartheta}$ i vektor opažanja $\boldsymbol{y}$ računamo kao produkt nezavisnih pojedinačnih vjerojatnosti.

Procjena parametara u slučaju poznatog pridruživanja opažanja

Pretpostavimo sad da su parametri komponenata mješavine kao i vektor težina $\boldsymbol{\eta}$ nepoznati. U ovoj cjelini procjenjujemo parametre konačne mješavine (1.33) za slučaj kada uz proizvoljno opažanje $\boldsymbol{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$ slučajnog vektora $\boldsymbol{Y}$, opažamo i odgovarajuće pridruživanje $\boldsymbol{s} = (s_1, \ldots, s_N)$. Procjenu parametara provodimo na potpunom skupu podataka $\{\boldsymbol{y}, \boldsymbol{s}\}$ i to dvjema standardnim metodama: metodom maksimizacije vjerodostojnosti i Bayesovskim zaključivanjem. Obje metode su temeljene na računu s ***funkcijom potpune vjerodostojnosti***[20] $L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta})$ koja odgovara uzoračkoj distribuciji potpunog skupa podataka uvjetno na nepoznati parametar $\boldsymbol{\vartheta}$. Vjerojatnost $L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta})$ izvodimo koristeći hijerarhijski model latentne varijable uveden u potpoglavlju 1.2 i to iz jednadžbi (1.29) i (1.31). Tražena vjerojatnost je onda

$$L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta}) = f(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{\vartheta}) f(\boldsymbol{s} \mid \boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(\boldsymbol{y_i} \mid s_i, \boldsymbol{\vartheta}) f(s_i \mid \boldsymbol{\vartheta}). \tag{1.43}$$

Primijetimo da je $f(\boldsymbol{y_i} \mid s_i = k, \boldsymbol{\vartheta}) = f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k)$ i $f(s_i = k \mid \boldsymbol{\vartheta}) = \eta_k$. Stavimo još da je $N_k(\boldsymbol{s}) = \#\{s_i = k\}$ broj opažanja pridruženih komponenti k. Uvrstimo spomenute vjerojatnosti i oznake u izvod potpune vjerodostojnosti pa dobijemo

[20] eng. *complete data likelihood*

$$L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \Big(f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \eta_k \Big)^{\mathbb{1}_{\{s_i=k\}}} = \prod_{k=1}^{K} \left(\prod_{i:\, S_i=k} f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \right) \left(\prod_{k=1}^{K} \eta_k^{N_k(\boldsymbol{s})} \right). \tag{1.44}$$

Provedimo sad procjenu parametara funkcije (1.44) metodom maksimalne vjerodostojnosti i Bayesovom metodom.

Procjena parametara ML metodom na potpunom skupu podataka

Funkcija potpune vjerodostojnosti izvedena u (1.44) sastoji se od $K + 1$ uzastopnih produkata faktoriziranih po parametrima. Prvih K faktora ovise samo o parametru $\boldsymbol{\theta_k}$ komponente $\mathcal{K}_k$, a posljednji ovisi samo o vektoru težina $\boldsymbol{\eta}$. Takva faktorizacija olakšava određivanje parametara modela jer omogućava da procjeni svakog parametra pristupamo izravno i neovisno od ostalih parametara. Kako bismo dodatno olakšali račun procjene, logaritmirat ćemo funkciju (1.44) pa koristiti metodu maksimalne log–vjerodostojnosti potpunih podataka. Dobivenu funkciju zatim maksimiziramo po vektoru parametara $\boldsymbol{\vartheta}$. Uvedimo oznaku indikatora pridruživanja $S_{ik} = \mathbb{1}_{\{s_i=k\}}$ i izvedimo potpunu log–vjerodostojnost prema navedenoj tehnici.

$$\begin{aligned} \log L^c(\boldsymbol{y}, \boldsymbol{S} \mid \boldsymbol{\vartheta}) &= \sum_{i=1}^{N} \sum_{k=1}^{K} S_{ik} \log \Big(f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k) \eta_k{}^{N_k(\boldsymbol{s})} \Big) = \sum_{i=1}^{N} \sum_{k=1}^{K} S_{ik} \Big(\log \left(f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) + \log \left(\eta_k{}^{N_k(\boldsymbol{s})} \right) \right) \\ &= \sum_{k=1}^{K} \sum_{i:\, s_i=k} \log \left(f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \right) + \sum_{k=1}^{K} N_k(\boldsymbol{s}) \log \eta_k. \end{aligned} \tag{1.45}$$

Izvest ćemo procjenu parametra $\boldsymbol{\theta}_k$ i η_k tako da izraz (1.45) deriviramo po traženom parametru. Dobivene su jednadžbe

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}_k} \log L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta}) &= \frac{\partial}{\partial \boldsymbol{\theta_k}} \sum_{j=1}^{K} \sum_{i:\, s_i=j} \log \Big(f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_j) \Big) = \frac{\partial}{\partial \boldsymbol{\theta_k}} \sum_{i:\, s_i=k} \log \left(f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \right) \\ &= \sum_{i:\, s_i=k} \frac{\frac{\partial}{\partial \boldsymbol{\theta}_k} f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k)}{f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k)} \end{aligned} \tag{1.46}$$

te

$$\frac{\partial}{\partial \eta_k} \log L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta}) = \frac{\partial}{\partial \eta_k} \sum_{j=1}^{K} N_j(\boldsymbol{s}) \log \Big(\eta_j \Big) = \frac{\partial}{\partial \eta_k} N_k(\boldsymbol{s}) \log(\eta_k) = \frac{N_k(\boldsymbol{s})}{\eta_k}. \tag{1.47}$$

Primijetimo da je $N = N_1(\boldsymbol{s}) + \cdots + N_K(\boldsymbol{s})$ i da procjena komponenata vektora težina $\boldsymbol{\eta}$ ovisi samo o broju opažanja pridruženih svakoj komponenti. Sada je procjenitelj komponente težine maksimalne log–vjerodostojnosti potpunog skupa podataka

$$\hat{\eta}_k = \frac{N_k(\boldsymbol{s})}{N}, \quad \forall k \in \mathcal{I}. \tag{1.48}$$

Svaki parametar $\boldsymbol{\theta}_k$ k–te komponente procjenjujemo iz opažanja pridruženih komponenti $\mathcal{K}_k$, za sve $k \in \mathcal{I}$. U slučaju kada komponente pripadaju nekoj Gaussovoj višedimenzionalnoj distribuciji, procjenitelji parametara maksimalne log–vjerodostojnosti potpunog skupa podataka su uzoračka aritmetička sredina i uzoračka kovarijacijska matrica

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N_k(\boldsymbol{s})} \sum_{i:\ s_i=k} \boldsymbol{y}_i\ , \tag{1.49}$$

$$\hat{\Sigma}_k = \frac{1}{N_k(\boldsymbol{s})} \sum_{i:\ s_i=k} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)^\tau (\boldsymbol{y}_i - \boldsymbol{\mu}_k). \tag{1.50}$$

Ako je broj opažanja pridruženih nekoj komponenti jako mali, može se dogoditi da MLE procjenitelj ne postoji (ili je degeneriran) ili da leži u rubu parametarskog prostora $\boldsymbol{\Theta}$. U slučaju da je komponenta $\mathcal{K}_k$ prazna, a budući da smo pretpostavili postojanje K podpopulacija, procjenitelj komponente težine η_k leži na rubu parametarskog prostora. Tada nemamo zadovoljene uvjete regularnosti pa nismo niti smjeli provoditi procjenu metodom maksimalne vjerodostojnosti. Kod modeliranja stvarnih podataka je zato važno što bolje ocijeniti mogući broj komponenata te, prije provođenja procjene ML metodom, provjeriti je li neka komponenta zapravo prazna. Najbolje procjene dobiju se ako sve komponente imaju *velik* broj pridruženih opažanja.

Bayesova procjena parametara na potpunom skupu podataka

Kako bismo odredili aposteriornu gustoću $f(\boldsymbol{\vartheta} \mid \boldsymbol{y}, \boldsymbol{s})$, primijenit ćemo Bayesov teorem 1.1.18 na apriornu gustoću $f(\boldsymbol{\vartheta})$ i funkciju potpune vjerodostojnosti $L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta})$. Relacija proporcionalnosti koju koristimo je

$$f(\boldsymbol{\vartheta} \mid \boldsymbol{s}, \boldsymbol{y}) \propto L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta}) f(\boldsymbol{\vartheta}). \tag{1.51}$$

Faktorizacijom potpune vjerodostojnosti (1.44), dobili smo $K + 1$ umnožaka. Na sličan način raspišemo apriornu gustoću pa dobijemo

$$f(\boldsymbol{\vartheta}) = f(\boldsymbol{\eta}) \prod_{k=1}^{K} f(\boldsymbol{\theta}_k). \tag{1.52}$$

Faktoriziranu apriornu gustoću uvrstit ćemo u relaciju Bayesove proporcionalnosti (1.51). Znamo da faktorizacijom po parametrima aposteriornu gustoću potpunih podataka možemo izraziti s

$$f(\boldsymbol{\vartheta} \mid \boldsymbol{s}, \boldsymbol{y}) = \prod_{k=1}^{K} f(\boldsymbol{\eta} \mid \boldsymbol{s}) f(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \boldsymbol{s}). \tag{1.53}$$

Faktore iz jednadžbe (1.53) možemo neovisno analizirati po parametrima $\boldsymbol{\theta}$, odnosno $\boldsymbol{\eta}$ iz sljedećih relacija:

$$f(\boldsymbol{\eta} \mid \boldsymbol{s}) \propto \prod_{k=1}^{K} f(\boldsymbol{\eta}) \eta_k{}^{N_k(\boldsymbol{s})}, \tag{1.54}$$

odnosno

$$f(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \boldsymbol{s}) \propto \prod_{i:\, S_i = k} f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k) f(\boldsymbol{\theta}_k). \tag{1.55}$$

Ovakva faktorizacija pogodna je za procjenu parametara mješavine jer, kao i u slučaju procjene korištenjem ML metode, procjeni svakog parametra pristupamo izravno. Dobivene izraze koristimo u metodi Bayesove procjene parametara opisane u potpoglavlju 1.1.

Procjena parametara u slučaju nepoznatog pridruživanja opažanja

U posljednjem dijelu potpoglavlja 1.3 razmatramo slučaj procjene parametara mješavine kada je poznat broj komponenata K, ali kada apriorno nemamo informaciju o realizaciji pridruživanja. Budući da parametre procjenjujemo na nepotpunom skupu podataka, a nije zanemariv utjecaj koji pridruživanje opažanja komponentama ima na zaključivanje, potrebne su složenije tehnike procjene. Ovaj problem je stoga daleko kompliciraniji od prethodnih. Razlog tomu je što, čak u slučaju da komponente mješavine pripadaju distribucijama koje se općenito lako modeliraju, ne postoje izravne statističke metode, već su potrebne metode koje koriste računalne algoritme za procjenu parametara. Jednom kada su parametri mješavine procijenjeni, koristimo zaključivanje s početka ovog potpoglavlja kako bismo pridružili opažanja odgovarajućim komponentama.

Neka je $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ vektor opažanja slučajne varijable $\boldsymbol{Y}$ iz konačne distribucije (1.33) te neka je $\boldsymbol{\vartheta} = (\boldsymbol{\theta}, \boldsymbol{\eta})$ vektor parametara koji procjenjujemo. Metoda momenata je početkom 20. stoljeća bila najčešće korištena metoda procjene parametara. Čuveni engleski matematičar Karl Pearson koristi ju još 1894. u svome radu *Contributions to the Mathematical Theory of Evolution* [11]. Pearson je modelirao binormalnu populaciju rakova kojoj je procijenio parametre distribucije. Jedan je od prvih autora znanstvenog članka na temu miješanih gustoća. Navodimo princip korištenja metode momenata u procjeni parametara.

Metoda momenata na nepotpunom skupu podataka
Neka je $H_j(\boldsymbol{Y})$ neka funkcija[21] slučajne varijable $\boldsymbol{Y}$ koju koristimo u procjeni parametra $\boldsymbol{\vartheta}_j$, za $j \in \{1, \ldots, d\}$, gdje je $d = \dim \boldsymbol{\vartheta}$. Uvjetno očekivanje računamo formulom

$$\mathbb{E}\left[H_j(\boldsymbol{Y} \mid \boldsymbol{\vartheta})\right] = \int_{\mathcal{Y}} H_j(\boldsymbol{y}) f(\boldsymbol{y} \mid \boldsymbol{\vartheta}) d\boldsymbol{y}. \tag{1.56}$$

Želimo odrediti parametre $\boldsymbol{\vartheta}$ tako da teoretski momenti $\mathbb{E}\left[H_j(\boldsymbol{Y} \mid \boldsymbol{\vartheta})\right]$ odgovaraju uzoračkim $\mathbb{E}\left[H_j^*(\boldsymbol{Y} \mid \boldsymbol{\vartheta})\right]$. Tražena jednakost je funkcija nepoznatog parametra

$$H_j^* = \sum_{k=1}^{K} \mathbb{E}\left[H_j(\boldsymbol{Y} \mid \boldsymbol{\theta}_k)\eta_k\right]. \tag{1.57}$$

Vektor parametara $\boldsymbol{\vartheta}$ onda odredimo kao rješenje sustava jednadžbi pa je za postojanje jedinstvenog rješenja nužno da imamo d linearno nezavisnih jednadžbi.

Procjena parametara ML metodom na nepotpunom skupu podataka
Razvojem računala i numeričkih metoda u drugoj polovici 20. stoljeća, iterativne metode potiskuju teorijske. Tako je fokus s metode momenata prebačen na već spomenutu metodu maksimalne vjerodostojnosti. Izvest ćemo ***vjerodostojnost mješavine*** $L(\boldsymbol{y} \mid \boldsymbol{\vartheta})$ kao zajedničku distribuciju opažanja $\boldsymbol{y_1}, \ldots, \boldsymbol{y}_N$ uvjetno na vektor parametara $\boldsymbol{\vartheta}$. Uvrstimo u definiciju vjerodostojnosti (1.10) raspis po komponentama mješavine:

$$L(\boldsymbol{y} \mid \boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(\boldsymbol{y}_i \mid \boldsymbol{\vartheta}) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k)\eta_k \right). \tag{1.58}$$

Primijetimo da se funkcija vjerodostojnosti mješavine sastoji od N faktora, od kojih je svaki sadrži sumu K pribrojnika. Usporedbe radi, funkciju potpune vjerodostojnosti mogli smo raščlaniti na $K + 1$ faktor te smo procjeni svakog parametra komponente pristupali izravno, a to u ovom slučaju nećemo moći. Formulu (1.58) možemo rastaviti u zbroj od K^N pribrojnika, ali ju ne možemo značajnije pojednostaviti. Slijede standardni postupci u procjeni parametara ML metodom: logaritmiranje i deriviranje izraza (1.58). Zbog velikog broja operacija koje smo do sad napravili, vidimo da je dobar smjer u daljnjoj analizi mješavine korištenje računalnih algoritama. Algoritam koji pritom koristimo je već ranije spomenuti EM algoritam. Ovdje započetu procjenu parametara mješavine u slučaju nepoznatog vektora pridruživanja dovršit ćemo u poglavlju 2 nakon što objasnimo EM algoritam.

[21] Primjerice $H_j(\boldsymbol{Y}) = \boldsymbol{Y}$ ili u slučaju računa m–tog momenta: $H_j(\boldsymbol{Y}) = (\boldsymbol{Y} - \boldsymbol{\mu})^m$.

Poglavlje 2

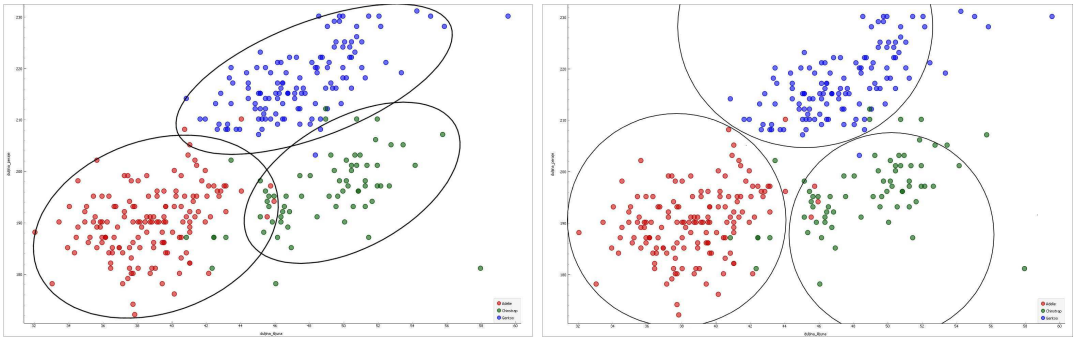
Algoritam maksimizacije očekivanja

Problem modeliranja konačnih mješavina dugi se niz godina rješavao metodama klasičnog statističkog zaključivanja koje smo objasnili u poglavlju 1. Preokret u pristupu procjeni parametara dogodio se uvođenjem računala u proces analize podataka. Razvojem numeričkih metoda u računarstvu, nametnula se ideja da metodu maksimalne vjerodostojnosti prilagodimo implementaciji na računalu. Formaliziranjem postupka procjene parametara u konačnici nastaje algoritam maksimizacije očekivanja. EM Algoritam su 1977. godine predstavili Dempster, Laird i Rubin[1] u radu *Maximum Likelihood from Incomplete Data via the EM Algorithm* [1], gdje su predložili postupak koji se sastoji od dva naizmjenično ponavljajuća koraka. Prema operacijama koje se izvode u svakom koraku, Dempster et al. su i iskovali naziv algoritma. Prvi je korak očekivanja (**E–korak**) u kojemu se, za trenutni procjenitelj parametara $\hat{\boldsymbol{\vartheta}}$, računa očekivanje potpune log–vjerodostojnosti izvedene u izrazu (1.45). Korak maksimizacije (**M–korak**) određuje novi procjenitelj parametara koji očekivanje iz E–koraka maksimizira. U svakom se koraku algoritma približavamo lokalnom maksimumu funkcije log–vjerodostojnosti jer algoritam nužno konvergira u smislu da niz procjenitelja parametara dobivenih u iteracijama algoritma konvergira prema stvarnoj vrijednosti parametara mješavine. Uz blage zahtjeve regularnosti na funkciju L^c, osiguravamo nužnu konvergenciju niza procjenitelja u njezin lokalni maksimum. Međutim, postizanje konvergencije može biti vrlo sporo, osobito ako početna vrijednost vektora parametara nije izabrana pažljivo. Algoritme grupiranja dijelimo prema načinu pridruživanja podataka komponentama na klase tvrdog i mekog grupiranja.[2] Algoritmi tvrdog grupiranja ne dozvoljavaju preklapanje grupa i dodjeljivanje oznaka više grupa jednoj jedinki. S druge strane, algoritmi mekog grupiranja ne određuju striktne granice između komponenata te računaju vjerojatnost da jedinka pripada svakoj od komponenata te pridružuju na temelju najvjerojatnijeg događaja. EM algoritam je stoga tipičan primjer algoritma mekog

[1] *Dempster et al.*

[2] eng. *hard and soft clustering*

grupiranja. Međutim, za određivanje inicijalnih vrijednosti parametara, dobar je izbor neki od algoritama tvrdog grupiranja, osobito neka inačica algoritma k–srednjih vrijednosti[3]. Komponente dobivene algoritmom k–srednjih vrijednosti su, u slučaju da mješavina pripada eksponencijalnoj familiji distribucija, disjunktne kugle u prostoru podataka. Grupe određene EM algoritmom često se preklapaju, te čine oblik elipsoida kao što je prikazano na slici 2.1.



Slika 2.1: meko i tvrdo grupiranje

Bojom su označene tri komponente kojima pripadaju podaci. Lijevi dijagram rasipanja prikazuje raspored grupa dobivenih mekim grupiranjem, dok desni prikazuje razmještaj dobiven tvrdim grupiranjem. Prikazana je prostorna raspršenost populacije koju modeliramo u potpoglavlju 2.2.

Osim procijenjenih parametara konačne mješavine, ovaj algoritam daje informaciju kojoj je komponenti pridruženo koje opažanje. Na konačne mješavine zato često indirektno nailazimo u brojnim znanstvenim disciplinama. Tako u biomedicini rješavamo problem klasifikacije genetskih promjena u organizmu ili određujemo značajke neke populacije živih bića. U poslovnoj ekonomiji postoji potreba za segmentacijom tržišta i kupaca te se prirodno nameće model miješanih gustoća. S gledišta strojnog učenja, EM algoritam pripada generativnim algoritmima te se proučava iz konteksta optimizacije učenja sistema. Osim navedenih područja, mnoge druge prirodne i društvene znanosti bave se modeliranjem konačnih mješavina u nekom njima svojstvenom kontekstu. Puno više o praktičnoj primjeni algoritma možemo pročitati u [4]. Frühwirth-Schnatter daje u *7.* poglavlju navedene knjige pregled različitih problema analize podataka koji koriste modele konačnih mješavina te nas upućuje na znanstvene radove u kojima su pojedine teme detaljnije obrađene. U ovom nam poglavlju predstoji dovršiti statističko zaključivanje započeto u potpoglavlju 1.3. Potpoglavlje 2.1 bavi se formaliziranjem računarskog pristupa modeliranju konačnih mješavina, dok je potpoglavlje 2.2 rezervirano za analizu i modeliranje stvarnih podataka.

[3] eng. *k–means algorithm*

2.1 Algoritam i procjena parametara

Promatramo populaciju neke Gaussove konačne mješavine koja se sastoji od $K \in \mathbb{N}$ različitih normalno distribuiranih komponenata. Kao u definiciji (1.26) mješavina ima $r \in \mathbb{N}$ značajki koje promatramo unutar slučajnog vektora $\boldsymbol{Y} = (Y_1, \ldots, Y_r)$. Iz mješavine je izabran proizvoljan uzorak $\boldsymbol{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_N})$ koji se sastoji od $N \in \mathbb{N}$ nezavisnih opažanja. Na temelju tih opažanja želimo izvesti zaključke o čitavoj populaciji. Problem koji rješavamo je određivanje parametara distribucije za svaku od K komponenata u slučaju kada vektor pridruživanja promatranog opažanja nije unaprijed poznat. Procjenu parametara provodimo ML metodom, a pritom rješavamo i problem pridruživanja opažanja komponentama. Prisjetimo se funkcije gustoće promatrane Gaussove mješavine:

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \sum_{k=1}^{K} \eta_k \left((2\pi)^{-\frac{r}{2}} (\det \Sigma_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{y} - \boldsymbol{\mu_k}) \Sigma_k^{-1} (\boldsymbol{y} - \boldsymbol{\mu_k})^{\tau} \right) \right), \ \forall \boldsymbol{y} \in \boldsymbol{Y}, \tag{2.1}$$

ML metoda procjene parametara određuje procjenitelj parametara koji maksimizira vjerodostojnost. Funkciju nepotpune vjerodostojnosti (1.58) logaritmiramo pa je nepotpuna log–vjerodostojnost

$$\log L(\boldsymbol{y} \mid \boldsymbol{\vartheta}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \eta_k f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \right). \tag{2.2}$$

Funkcija gustoće komponente $\mathcal{K}_k$ uvjetno na odgovarajući parametar $\boldsymbol{\theta}_k$ je Gaussova komponentna gustoća koju vidimo u gustoći mješavine (2.1).

EM algoritam

Algoritam maksimizacije očekivanja je ponavljajući algoritam koji parametre mješavine procjenjuje maksimizirajući očekivanje potpune log–vjerodostojnosti. Budući da od ranije nemamo informacije o latentnim varijablama, potpunu log–vjerodostojnost ćemo maksimizirati u terminima njezinog uvjetnog očekivanja uz dani poznati vektor opažanja $\boldsymbol{y}$ i trenutnu vrijednost parametara $\boldsymbol{\vartheta}^{(w)}$. Indeks $w \in \mathbb{N}_0$ označava da provodimo $(w+1)$–u iteraciju algoritma. U potpoglavlju 2.8. knjige *Finite Mixture Models* [8] detaljno je objašnjena primjena algoritma na modele konačnih mješavina te je opravdano korištenje uvjetnog očekivanja potpune log–vjerodostojnosti. To ćemo zaključivanje izvesti pri definiranju E–koraka. Označimo s $\boldsymbol{\vartheta}^{\star}$ stvarnu vrijednost vektora parametara koji maksimizira potpunu log–vjerodostojnost. Algoritam započinje inicijalizacijom vektora parametara $\boldsymbol{\vartheta}^{(\mathbf{0})}$. Izbor prvih vrijednosti parametara može biti slučajan, ali je bolje napraviti analizu funkcije $\log L^c(\boldsymbol{y}, \boldsymbol{s} \mid \boldsymbol{\vartheta})$ i odabrati vrijednost koja je relativno blizu stvarne vrijednosti $\boldsymbol{\vartheta}^{\star}$. Također,

u slučaju da log–vjerodostojnost ima višestruke nultočke, algoritam bi trebalo provesti na širokom izboru početnih vrijednosti.

E–korak

Uvedimo oznaku za uvjetno očekivanje potpune log–vjerodostojnosti uz dani poznati vektor opažanja $\boldsymbol{y}$ i trenutnu vrijednost parametara $\boldsymbol{\vartheta}^{(w)}$ u $(w+1)$–oj iteraciji:

$$Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(w)}) = \mathbb{E}_{\boldsymbol{\vartheta}^{(w)}} \left[\log L^c(\boldsymbol{\vartheta}, \boldsymbol{s}) \mid \boldsymbol{y} \right]. \tag{2.3}$$

U jednadžbi (2.3) oznaka $\boldsymbol{\vartheta}^{(w)}$ označava da u $(w+1)$–om koraku na očekivanje uvjetuje i trenutna vrijednost procijenjenog vektora parametara. Kao što je najavljeno, računamo uvjetno očekivanje potpune log–vjerodostojnosti iako smo u slučaju kada je pridruživanje $\boldsymbol{S}$ apriorno nepoznato. Primijetimo da je funkcija potpune log–vjerodostojnosti $\log L^c$ linearna s obzirom na nepoznatu realizaciju pridruživanja. Računamo uvjetno očekivanje pridruživanja uz dano opažanje i trenutnu vrijednost parametara:

$$\mathbb{E}_{\boldsymbol{\vartheta}^{(w)}} \left[S_i = s_k \mid \boldsymbol{y} \right] = \mathbb{P}_{\boldsymbol{\vartheta}^{(w)}}(s_{ik} = 1 \mid \boldsymbol{y}), \quad \forall i \in \{1, \ldots, N\}, \forall k \in \mathcal{I}, \tag{2.4}$$

gdje je $\mathcal{I}$ skup indikatora komponenata. Primijetimo da smo dobili aposteriornu vjerojatnost pridruživanja i–tog opažanja komponenti $\mathcal{K}_k$. Navedena vjerojatnost je odgovornost h_{ik} uvedena izrazom (1.36). Uvrstimo u izraz uvjetnog očekivanja potpune log–vjerodostojnosti (2.3) odgovornost iz (2.4) pa dobijemo

$$Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(w)}) = \sum_{i=1}^{N} \sum_{k=1}^{K} h_{ik} \left(\log \eta_k + \log f_k(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k) \right). \tag{2.5}$$

Dobivenu funkciju maksimiziramo u sljedećem koraku. Budući da su parametri težina i komponenata odvojeni operacijom zbrajanja, njihovoj procjeni pristupamo neovisno jer deriviranjem jednadžbe (2.5) po nekom određenom parametru, ostaje samo izraz s tim parametrom.

M–korak

Algoritam je još uvijek u $(w+1)$–oj iteraciji i potrebno je odrediti novu vrijednost vektora parametara za koju funkcija (2.5) postiže maksimum. Izvedimo prvo procjenitelj vektora težina. Na potpunom skupu podataka izveli smo procjenu parametara težina u jednadžbi (1.48) kao količnik ukupnog broja opažanja pridruženih promatranoj komponenti $N_k(\boldsymbol{s})$ i ukupnog broja opažanja N. Sada međutim, radimo s nepotpunim skupom podataka i ne znamo koja je realizacija pridruživanja. Zbog konstrukcije E–koraka, ideja je zamijeniti broj opažanja $N_k(\boldsymbol{s})$ s uvjetnim očekivanjem h_{ik} uz trenutne vrijednosti parametara. Nova vrijednost procjenitelja parametara težine je onda

$$\eta_k{}^{(w+1)} = \frac{1}{N}\sum_{i=1}^{n} h_{ik}^{(w)}, \quad \forall k \in \mathcal{I}. \tag{2.6}$$

Utjecaj svakog od opažanja $\boldsymbol{y}_i$ na procjenu težine komponente η_k nije zanemariv i on odgovara trenutnoj vrijednosti aposteriorne vjerojatnosti $h_{ik}^{(w)}$. Zbog normiranosti vektora težina (1.25), dovoljno je procijeniti $K-1$ parametar težine u trenutnoj iteraciji, a jednu težinu odrediti pomoću ostalih. Izbor težine komponente koju ne procjenjujemo već računamo koristeći normiranost je proizvoljan. Ovdje je izbor pao na procjenitelj težine $\eta_K^{(w+1)}$:

$$\eta_K{}^{(w+1)} = 1 - \sum_{k=1}^{K-1} \eta_k{}^{(w+1)}. \tag{2.7}$$

Preostalo nam je još odrediti procjenitelje distribucijskih parametara komponenata u $(w+1)$–oj iteraciji koristeći ML metodu procjene na uvjetnom očekivanju (2.5). Za svaku ćemo komponentu odrediti parametre koji maksimiziraju uvjetno očekivanje tako da odredimo rješenje diferencijalne jednadžbe

$$\frac{\partial}{\partial \boldsymbol{\theta}_k} Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(w)}) = \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij} \frac{\partial\left(\log f_j(\boldsymbol{y}_i \mid \boldsymbol{\theta}_j)\right)}{\partial \boldsymbol{\theta}_k} = 0. \tag{2.8}$$

Izvedimo log–vjerodostojnost k–te komponente Gaussove mješavine kako bismo mogli odrediti procjenitelje parametara komponenata mješavine

$$\begin{aligned}\log f_k(\boldsymbol{y} \mid (\boldsymbol{\mu}_k, \Sigma_k)) &= \log\left(\frac{\exp\left\{\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_k)\Sigma_k{}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_k)^\tau\right)\right\}}{(2\pi)^{\frac{r}{2}}(\det\Sigma_k)^{\frac{1}{2}}}\right) = \\ &= -\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_k)\Sigma_k{}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_k)^\tau - \log\left((2\pi)^{\frac{r}{2}}(\det\Sigma_k)^{\frac{1}{2}}\right).\end{aligned} \tag{2.9}$$

Procjenitelje očekivanja i kovarijacijske matrice komponente $\mathcal{K}_k$ određujemo rješavajući sljedeće jednadžbe:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}_k} Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(w)}) &= \sum_{i=1}^{N} h_{ik} \frac{\partial\Big(\log f_k(\boldsymbol{y}_i \mid (\boldsymbol{\mu}_k, \Sigma_k))\Big)}{\partial \boldsymbol{\mu}_k} = \\ &= \sum_{i=1}^{N} h_{ik} \frac{\partial\left(-\frac{1}{2}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)\Sigma_k{}^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)^\tau - \log\left((2\pi)^{\frac{r}{2}}(\det\Sigma_k)^{\frac{1}{2}}\right)\right)}{\partial \boldsymbol{\mu}_k} \\ &= \sum_{i=1}^{N} h_{ik} \frac{\partial\left(-\frac{1}{2}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)\Sigma_k{}^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)^\tau\right)}{\partial \boldsymbol{\mu}_k} = 0,\end{aligned} \tag{2.10}$$

$$\frac{\partial}{\partial \Sigma_k} Q(\boldsymbol{\vartheta} \mid \boldsymbol{\vartheta}^{(w)}) = \sum_{i=1}^{N} h_{ik} \frac{\partial \left(\log f_k(\boldsymbol{y}_i \mid (\boldsymbol{\mu}_k, \Sigma_k))\right)}{\partial \Sigma_k} =$$

$$= \sum_{i=1}^{N} h_{ik} \frac{\partial \left(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}_k)\Sigma_k{}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^\tau - \log\left((2\pi)^{\frac{r}{2}}(\det \Sigma_k)^{\frac{1}{2}}\right)\right)}{\partial \Sigma_k} = 0. \quad (2.11)$$

Parametre koje anuliraju jednadžbe (2.10) i (2.11) možemo zapisati u zatvorenoj formi:

$$\boldsymbol{\mu}_k{}^{(w+1)} = \frac{\sum_{i=1}^{N} h_{ik}^{(w)} \boldsymbol{y}_i}{\sum_{i=1}^{N} h_{ik}^{(w)}}, \quad (2.12)$$

$$\Sigma_k{}^{(w+1)} = \frac{\sum_{i=1}^{N} h_{ik}^{(w)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_i^{(w+1)}\right)\left(\boldsymbol{y}_i - \boldsymbol{\mu}_i^{(w+1)}\right)^\tau}{\sum_{i=1}^{N} h_{ik}^{(w)}}. \quad (2.13)$$

Distribucijama koje pripadaju eksponencijalnoj familiji distribucija je zbog forme njihove funkcije gustoće jednostavno provesti procjenu parametara. Vratimo li se na izvod nepotpune vjerodostojnosti mješavine u jednadžbi (1.58), vidimo da bi direktnu primjenu ML metode bilo teško egzaktno izvesti. S druge strane, procjenitelje dobivene EM–algoritmom dobili smo nizom jednostavnih operacija. Račun je osobito jednostavan ako ga provodimo na računalu, ali je i tada korisno izbaciti redundantne operacije koje provode iste izračune kroz sve iteracije. Broj računskih operacija ćemo optimizirati uvođenjem dovoljnih statistika za vektor parametara $\boldsymbol{\vartheta}$. Jedan od prijedloga smanjenja broja operacija opisan je u potpoglavlju *3.2* knjige [8] gdje su uvedene statistike

$$\begin{aligned} T_{k1}^{(w)} &= \sum_{i=1}^{N} h_{ik}^{(w)}, \\ T_{k2}^{(w)} &= \sum_{i=1}^{N} h_{ik}^{(w)} \boldsymbol{y}_i \quad \text{i} \\ T_{k3}^{(w)} &= \sum_{i=1}^{N} h_{ik}^{(w)} \boldsymbol{y}_i \boldsymbol{y}_i{}^\tau, \end{aligned} \quad (2.14)$$

za svaku komponentu $\mathcal{K}_k$, $k \in \mathcal{I}$. Sada kovarijacijsku matricu k–te komponente u $(w+1)$–oj iteraciji možemo odrediti po formuli

$$\Sigma_k{}^{(w+1)} = \frac{T_{k3}^{(w)} - \left(T_{k1}^{(w)}\right)^{-1} T_{k2}^{(w)} \left(T_{k2}^{(w)}\right)^\tau}{T_{k1}^{(w)}}. \quad (2.15)$$

Korištenjem formule (2.15) umjesto formule (2.13) za procjenu kovarijacijske matrice komponente $\mathcal{K}_k$, za 50% smanjujemo CPU vrijeme potrebno za izvođenje iteracije.[4] Koraci E i M algoritma naizmjenično se ponavljaju do postizanja konvergencije. Potrebno je stoga zadati uvjet zaustavljanja prije no što pokrenemo algoritam. Tipičan uvjet zaustavljanja je kada je razlika vektora dobivenih uzastopnim iteracijama ili razlika njihovih log–vjerodostojnosti dovoljno mala. Dempster et al. su prvi diskutirali konvergenciju EM algoritma, ali se u formalnom dokazu predstavljenom u njihovom članku [1] potkrala pogreška. Na pogrešku je ukazao Jeff C. F. Wu u svom članku *On the Convergence Properties of EM Algorithm* [17] gdje je i dao dokaz konvergencije algoritma u seriji vezanih teorema.

Konvergencija EM algoritma

Iskazat ćemo glavni teorem konvergencije EM algoritma čiji dokaz možemo pronaći u [17]. Prije toga moramo još eksplicitno postaviti uvjete u kojima algoritam nužno konvergira. Na problem možemo naići u M–koraku ako rješenje diferencijalne jednadžbe (2.5) ne postoji u zatvorenoj formi. U tom su slučaju Dempster et al. predložili uvođenje preslikavanja $M\colon \boldsymbol{\Theta} \to \boldsymbol{\Theta}$ za koje vrijedi

$$\boldsymbol{\vartheta}^{(w+1)} = M\left(\boldsymbol{\vartheta}^{(w)}\right), \quad \forall w \in \mathbb{N}_0. \tag{2.16}$$

Ovako definirano preslikavanje koristimo u M–koraku za odabir novog procjenitelja vektora parametara, a ono spada u klasu preslikavanja ***točke–u–skup***.[5] Algoritam tada nazivamo ***generalizirani EM algoritam***, ili kraće **GEM algoritam**. Rezultat uvođenja preslikavanja (2.16) je da je niz $\left(L\left(\boldsymbol{\vartheta}^{(w+1)}\right)\right)_{w\in\mathbb{N}_0}$ neopadajući. U iskazu teorema koristit ćemo oznaku $\mathcal{M}$ za skup lokalnih maksimuma funkcije vjerodostojnosti, a oznaku $\mathcal{S}$ za skup stacionarnih točaka funkcije vjerodostojnosti na interioru prostora parametara.

Teorem 2.1.1. *Neka je $\left(\boldsymbol{\vartheta}^{(w+1)}\right)_{w\in\mathbb{N}_0}$ niz parametara određenih pomoću preslikavanja* (2.16) *u M–koraku GEM algoritma. Pretpostavimo da vrijedi:*

i. M je zatvoreno[6] preslikavanje točke–u–skup na komplementu skupa $\mathcal{S}$.

ii. $L\left(\boldsymbol{\vartheta}^{(w+1)}\right) > L\left(\boldsymbol{\vartheta}^{(w)}\right)$, za sve $\boldsymbol{\vartheta}^{(w)} \notin \mathcal{S}$.

Tada su sva gomilišta niza $(\boldsymbol{\vartheta}^{(w)})$ limesi i vrijedi da su stacionarne točke funkcije L. Također, niz $\left(L\left(\boldsymbol{\vartheta}^{(w)}\right)\right)_{w\in\mathbb{N}_0}$ monotono konvergira prema nekoj vrijednosti $\tilde{L} = L(\tilde{\boldsymbol{\vartheta}})$, gdje je $\tilde{\boldsymbol{\vartheta}} \in \mathcal{S}$.

[4] CPU vrijeme je vrijeme potrebno procesoru računala da izvrši zadane naredbe.

[5] eng. *point–to–set map* — u matematičkom programiranju ovakva preslikavanja definiraju algoritme, a iteracije algoritma generiraju niz $\boldsymbol{\vartheta}^{(w)}$ t.d. $\boldsymbol{\vartheta}^{(w+1)} \in M(\boldsymbol{\vartheta}^{(w)})$

[6] eng. *closed map* — zatvorena preslikavanja su ona koja preslikavaju zatvoreni skup u zatvoreni skup

Korolar 2.1.2. *Tvrdnja teorema 2.1.1 vrijedi i u slučaju kada tvrdnje (i.) i (ii.) vrijede na komplementu skupa $\mathcal{M}$. Tada limes niza vrijednosti funkcija log–vjerodostojnosti konvergira u lokalni maksimum, tj.*

$$\lim_{w\to\infty} L\left(\boldsymbol{\vartheta}^{(w)}\right) = L(\tilde{\boldsymbol{\vartheta}}), \qquad \textit{gdje je } \tilde{\boldsymbol{\vartheta}} \in \mathcal{M}. \tag{2.17}$$

Sad kad znamo da algoritam konvergira, zanima nas kojom brzinom. Brzinu konvergencije algoritma možemo opisati matrično, a pritom ćemo opet koristiti preslikavanje (2.16) uvedeno ranije. Ako niz $\boldsymbol{\vartheta}^{(w)}$ konvergira prema nekoj fiksnoj točki $\boldsymbol{\vartheta}^\circ$ i ako je preslikavanje M neprekidno, onda je $\boldsymbol{\vartheta}^\circ$ fiksna točka algoritma, tj. vrijedi

$$\boldsymbol{\vartheta}^\circ = M(\boldsymbol{\vartheta}^\circ). \tag{2.18}$$

Primijenimo li Taylorov razvoj preslikavanja M oko točke $\boldsymbol{\vartheta}^{(w)} = \boldsymbol{\vartheta}^\circ$, tada postoji okolina točke $\boldsymbol{\vartheta}^\circ$ na kojoj vrijedi

$$\boldsymbol{\vartheta}^{(w+1)} - \boldsymbol{\vartheta}^\circ \approx J(\boldsymbol{\vartheta}^\circ)\left(\boldsymbol{\vartheta}^{(w)} - \boldsymbol{\vartheta}^\circ\right), \tag{2.19}$$

gdje je $J(\boldsymbol{\vartheta})$ d–dimenzionalna Jacobijeva matrica preslikavanja $M(\boldsymbol{\vartheta}) = (M_1(\boldsymbol{\vartheta}), \ldots, M_d(\boldsymbol{\vartheta}))$. Stoga u okolini fiksne točke $\boldsymbol{\vartheta}^\circ$ EM algoritam konvergira linearno pa Jacobijevu matricu $J(\boldsymbol{\vartheta}^\circ)$ nazivamo ***matricom brzine konvergencije***.[7] Na prostoru parametara možemo definirati mjeru opažene konvergencije kao globalnu brzinu konvergencije sljedećim izrazom

$$r = \lim_{w\to\infty} \frac{\left\|\boldsymbol{\vartheta}^{(w+1)} - \boldsymbol{\vartheta}^\circ\right\|}{\left\|\boldsymbol{\vartheta}^{(w)} - \boldsymbol{\vartheta}^\circ\right\|}, \tag{2.20}$$

gdje je $\|\cdot\|$ proizvoljna norma na $\mathbb{R}^d$. Lako se pokaže da, uz određene uvjete regularnosti vrijedi da mjeru konvergencije možemo poistovjetiti s najvećom svojstvenom vrijednosti matrice $J(\boldsymbol{\vartheta}^\circ)$. U praksi ipak brzinu konvergencije računamo formulom

$$r = \lim_{w\to\infty} \frac{\left\|\boldsymbol{\vartheta}^{(w+1)} - \boldsymbol{\vartheta}^{(w)}\right\|}{\left\|\boldsymbol{\vartheta}^{(w)} - \boldsymbol{\vartheta}^{(w-1)}\right\|}. \tag{2.21}$$

U potpoglavlju 2.2 prezentirat ćemo pristup modeliranju Gaussovih mješavina stvarne populacije. Podatke ćemo analizirati pomoću metoda objašnjenih u poglavlju 1 i algoritma

[7] Ovaj pristup računanja brzine konvergencije objašnjen je u 2. poglavlju knjige Finite Mixture Models.

maksimizacije očekivanja pa ovdje navodimo njegov pseudo kod.

unos: Učitaj početnu vrijednost vektora parametara $\boldsymbol{\vartheta} \leftarrow (\eta_1, \ldots, \eta_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$
ispis: Spremi vektor procijenjenih parametara distribucije $\hat{\boldsymbol{\vartheta}} \leftarrow \boldsymbol{\vartheta}$

ponavljaj

E–korak: Izračunaj trenutnu odgovornost h_{ik}, $\forall i = 1, \ldots, N$, $\forall k \in \mathcal{I}$.

$$h_{ik} \leftarrow \frac{\mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = k, \boldsymbol{\vartheta})\eta_k}{\sum_{j=1}^{K} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y_i} \mid s_i = j, \boldsymbol{\vartheta})\eta_j}$$

M–korak: Izračunaj nove vrijednosti vektora parametara u ovisnosti o trenutnoj vrijednosti h_{ik}, $\forall k \in \mathcal{I}$.

$$\eta_k \leftarrow \frac{1}{N} \sum_{i=1}^{n} h_{ik}$$

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{i=1}^{N} h_{ik} \boldsymbol{y}_i}{\sum_{i=1}^{N} h_{ik}}$$

$$\Sigma_k \leftarrow \frac{\sum_{i=1}^{N} h_{ik} (\boldsymbol{y}_i - \boldsymbol{\mu}_i)(\boldsymbol{y}_i - \boldsymbol{\mu}_i)^\tau}{\sum_{i=1}^{N} h_{ik}}$$

Izračunaj novu vrijednost funkcije log–vjerodostojnosti.

$$\log L(\boldsymbol{y} \mid \boldsymbol{\vartheta}) \leftarrow \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \eta_k f(\boldsymbol{y_i} \mid \boldsymbol{\theta}_k) \right)$$

ako je zadovoljen uvjet konvergencije **onda**
    Izađi iz petlje.
inače
    Prijeđi na sljedeću iteraciju.
    $w \leftarrow w + 1$
kraj
dok ne bude zadovoljen uvjet konvergencije

Algoritam 1: EM algoritam

2.2 Praktična primjena algoritma

U ovom potpoglavlju ćemo primijeniti metode procjene parametara na stvarnim podacima te obraditi rezultate.[8] Modeliramo populaciju pingvina otočja Palmer na Antarktici prema podacima prikupljenima u sklopu istraživanja značajki triju vrsta pingvina Adélie, Chinstrap i Gentoo. Otočje se sastoji od otoka Torgersen, Biscoe i Dream, a podaci su prikupljani od 2007. do 2009. godine. Istraživači, na čelu s dr. Kristen Gorman, su bilježili mjere veličine, podatke o leglu, udio relevantnih izotopa u krvi, otok na kojemu je jedinka pronađena te pripadnost vrsti. Takvi podaci su primjer potpunog skupa podataka, ali ćemo ih modelirati pod pretpostavkom da unaprijed ne znamo kojoj vrsti pingvina određena jedinka pripada. Cilj nam je na temelju mjera veličine promatrane jedinke odrediti vrstu pingvina. Budući da se radi o životinjskoj populaciji, smisleno je da je distribucija svake komponente višedimenzionalna normalna distribucija. Za svaku ćemo vrstu pingvina odrediti parametre distribucije i pripadne proporcije miješanja. Nakon završetka modeliranja, usporedit ćemo pridruživanja dobivena EM algoritmom sa stvarnim podacima i izračunati pogrešku. Ovo je potpoglavlje podijeljeno u cjeline prema fazi modeliranja.

Pripremna obrada podataka

Koristimo tablicu penguins.csv koja sadrži kategoričke varijable (*vrsta, otok, spol, godina*) i numeričke variable (*duljina kljuna, visina kljuna, duljina peraje, masa tijela*). Strukturu podataka možemo vidjeti u tablici 2.1.

r.br.	vrsta	otok	duljina kljuna	visina kljuna	duljina peraje	masa tijela	spol	godina
20	Adelie	Torgersen	46	21.5	194	4200	m	2007
21	Adelie	Biscoe	37.8	18.3	174	3400	f	2007
66	Adelie	Biscoe	41.6	18	192	3950	m	2008
74	Adelie	Torgersen	45.8	18.9	197	4150	m	2008
155	Gentoo	Biscoe	48.7	14.1	210	4450	f	2007
189	Gentoo	Biscoe	42.6	13.7	213	4950	f	2008
277	Chinstrap	Dream	50	19.5	196	3900	m	2007
335	Chinstrap	Dream	45.6	19.4	194	3525	f	2009

Tablica 2.1: Tablični prikaz podataka o pingvinima

Prikazan je dio originalnih podataka pri čemu je zaglavlje tablice prevedeno na hrvatski jezik. Također, muške jedinke označavamo slovom **m**, a ženske slovom **f**.

[8] Podaci su preuzeti s GitHub stranice `https://github.com/allisonhorst/palmerpenguins` dr. Allison Horst 17. kolovoza 2020. te su javno dostupni prema *Creative Commons Zero v1.0 Universal* licenci.

Jedinke prikazane u tablici 2.1 su izabirane tako da reprezentiraju vrijednosti kategoričkih varijabli. Izmjerene duljine su navedene u milimetrima, a masa u gramima. Populaciju pingvina modeliramo prema numeričkim varijablama. Svaku jedinku stoga karakteriziramo 4–dimenzionalnim slučajnim vektorom $\boldsymbol{Y} = (Y_1, Y_2, Y_3, Y_4)$ čije komponente redom odgovaraju navedenim numeričkim varijablama. Prema notaciji iz definicije (1.2.1) stavimo $r = 4$. U tablici su zabilježene značajke o 344 jedinke pingvina, ali u analizu ne uvrštavamo one jedinke koje nemaju zabilježene sve numeričke varijable. Modeliranje stoga provodimo na populaciji koja se sastoji od 342 jedinke pa je $N = 342$. Također, iz informacija o istraživanju, znamo da se populacija pingvina otočja Palmer sastoji od tri vrste, stoga je $K = 3$. Da bi model bio dovoljno dobar, potrebno je prilagoditi broj opažanja promatranom broju značajki kako bi se izbjegla prenaučenost i podnaučenost modela. Često korištena smjernica o broju potrebnih opažanja u ovisnosti od broja promatranih značajki zahtijeva da su uređeni relacijom $2^r < N$. Ovaj zahtjev je u našem slučaju zadovoljen pa smatramo da imamo dovoljan broj podataka da model bude dobar.

Procjena parametara EM algoritmom

Računalni kôd je pisan u jeziku *Python* [15] i možemo ga pronaći u dodatku A. Svi grafički prikazi dani u ovom radu napravljeni su korištenjem Pythonovog paketa *matplotlib* i softvera *Orange* [2]. Prvi korak u modeliranju mješavine je inicijalizacija parametara, a ona je provedena algoritmom k–srednjih vrijednosti ++.[9] čiji je kôd također napisan u dodatku A.

Algoritam k–srednjih vrijednosti ++

Algoritam K++ je implementiran u skripti k_means_pp.py, gdje učitavamo originalnu tablicu penguins.csv. Nakon izvođenja algoritma, spremamo novu tablicu potpuni_podaci.csv dobivenu spajanjem originalne tablice i vektora pridruživanja. Ovaj se algoritam razlikuje od običnog algoritma k–srednjih vrijednosti samo u načinu izbora prvog ***centroida*** svake komponente. Centroid je središte mase komponente, a računamo ga kao srednju vrijednost svih podataka pridruženih toj komponenti. Prvi centroid biramo nasumično, dok ostalih $K - 1$ biramo tako da iz skupa preostalih jedinki izaberemo one koje su, po nekoj metrici prostora $\mathbb{R}^r$, najudaljenije od već izabranih centroida. Grupiranje sad nastavljamo običnim algoritmom k–srednjih vrijednosti koji podatke grupira izmjenjujući ***korak pridruživanja*** i ***korak određivanja novih centroida***. U koraku pridruživanja svakoj jedinki dodijelimo oznaku one komponente čiji je centroid njoj najbliži. Nakon što se sve jedinke pridruže nekoj komponenti, nove centroide računamo kao aritmetičku sredinu podataka po grupama, a algoritam se ponavlja do postizanja konvergencije. Po završetku provođenja algoritma, jedinke su pridružene komponentama i tada procjenjujemo distribucijske parametre pojedine

[9] U nastavku teksta koristimo skraćeni naziva algoritma — ***algoritam K++***.

vrste metodom momenata. Time je završen postupak inicijalizacije parametara mješavine i inicijalnog pridruživanja jedinki komponentama. Ovaj način grupiranja podataka spada u klasu tvrdih grupiranja jer je svaku jedinku moguće pridružiti točno jednoj komponenti. Iako su podaci pridruženi komponentama, a distribucijski parametri vrsta procijenjeni, algoritam K++ nije optimalan izbor za grupiranje konačnih mješavina. Razlog tomu je što komponente ne možemo striktno odvojiti što je upravo slučaj s populacijom pingvina koju modeliramo. Kad u istom biomu živi više genetski sličnih podvrsta iste vrste životinje ili biljke, često dolazi do miješanja među populacijama. Može se dogoditi da neku jedinku krivo pridružimo komponenti kojoj ne pripada, samo zato što joj je centroid te komponente najbliži. Zbog toga je bolje koristiti pristup koji za svaku komponentu računa vjerojatnost da joj jedinka pripada. Upravo je na tom principu koncipiran EM algoritam s kojim i nastavljamo modeliranje.

EM algoritam

Nakon provedene inicijalizacije parametara i pridruživanja, možemo krenuti s algoritmom maksimizacije očekivanja koji je detaljno objašnjen u potpoglavlju 2.1. Pokrećemo skriptu em_algoritam.py koja učitava tablicu potpuni_podaci.csv dobivenu korištenjem algoritma k–srednjih vrijednosti ++. Zadajemo uvjet zaustavljanja na način da se mora ispuniti barem jedna od dvije tvrdnje uvjeta.

- Razlika vektora parametara procijenjenih u uzastopnim iteracijama je dovoljno mala, tj. manja od $\delta = 10^{-7}$

$$\left|\boldsymbol{\vartheta}^{(w+1)} - \boldsymbol{\vartheta}^{(w)}\right| < 10^{-7}, \quad \text{za neki } w \in N_0$$

- Razlika vrijednosti funkcija log–vjerodostojnosti u uzastopnim iteracijama je dovoljno mala, tj. manja od $\varepsilon = 10^{-5}$

$$\left|L\left(\boldsymbol{\vartheta}^{(w+1)}\right) - L\left(\boldsymbol{\vartheta}^{(w)}\right)\right| < 10^{-5}, \quad \text{za neki } w \in N_0$$

Ako je za neki $w \in \mathbb{N}_0$ ispunjena jedna od navedenih tvrdnji onda smatramo da je algoritam iskonvergirao. Dobiveni vektor pridruživanja upisujemo u novu tablicu konacna_tablica.csv u kojoj su navedene vrijednosti pojedinih karakteristika jedinki, kao i pridruživanje ranije određeno algoritmom K++.

Analiza rezultata

Uz ovako definiran uvjet zaustavljanja, konvergencija se postiže nakon nekoliko desetaka iteracija EM algoritma. Uspoređujući rezultate dobivene pokretanjem računalnog kôda u

više navrata, primijetili smo da se procijenjeni parametri ne razlikuju značajno, tj. da dolaze iz $\varepsilon = 10^{-4}$ okoline stvarnog parametra. Razlika nastaje zbog slučajnog odabira prvog centroida u inicijalizaciji parametara algoritmom K++. Prikazat ćemo rezultate dobivene jednim pokretanjem algoritma te ih interpretirati. Konačna tablicu za jedinke predstavljene u tablici 2.1 je

r.br.	**vrsta**	**duljina kljuna**	**visina kljuna**	**duljina peraje**	**masa tijela**	**spol**	**godina**	**K++**	**EM**
20	Adelie	46	21.5	194	4200	m	2007	0	0
21	Adelie	37.8	18.3	174	3400	f	2007	1	0
66	Adelie	41.6	18	192	3950	m	2008	1	0
74	Adelie	45.8	18.9	197	4150	m	2008	0	1
155	Gentoo	48.7	14.1	210	4450	f	2007	0	2
189	Gentoo	42.6	13.7	213	4950	f	2008	2	2
277	Chinstrap	50	19.5	196	3900	m	2007	1	1
335	Chinstrap	45.6	19.4	194	3525	f	2009	1	1

Tablica 2.2: Tablični prikaz pridruživanja jedinki pingvina

U stupcu **K++** dane su oznake komponenata određene algoritmom K++, dok su u stupcu **EM** navedeni identifikatori komponenata određenih EM algoritmom.

Iz originalnih podataka o istraživanju znamo za svaku jedinku kojoj vrsti ona pripada pa možemo provjeriti točnost pridruživanja dobivenih EM algoritmom. Iz populacije 342 jedinke pingvina tek su četiri krivo pridružene, stoga je postotak točnog pridruživanja 98.83%. Radi se o dvije jedinke vrste Adelie koje su pridružene vrsti Chinstrap i o dvije jedinke vrste Chinstrap pridružene vrsti Adelie. To nam daje naslutiti da su diferencijacijska obilježja vrste Gentoo najizraženija. U tablici 2.2 vidimo da je jedinka vrste Adelie pod rednim brojem 74 pogrešno pridružena vrsti Chinstrap. Promjenom uvjeta zaustavljanja algoritma na način da smanjimo dozvoljenu razliku među parametrima, nismo izbjegli krivo pridruživanje za ove četiri jedinke. Potpuno točno pridruživanje dobili smo samo u slučaju kada za uvjet zaustavljanja zahtjevamo ispunjenje obje tvrdnje uvjeta. Tada je potrebno čak 30% više iteracija da se osigura konvergencija. Navodimo procijenjene distribucijske i težinske parametre komponenata miješane populacije pingvina te prikazujemo dijagram rasipanja po vrstama:

Adelie: $\eta_A = 0.446, \quad \boldsymbol{\mu}_A = (38.81,\ 18.32,\ 189.71,\ 3691.57)$

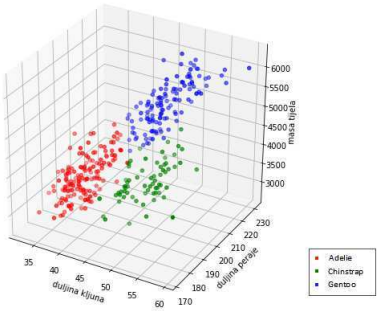
$$\Sigma_A = \begin{bmatrix} 7.00 & 1.17 & 4.46 & 622.21 \\ 1.17 & 1.49 & 2.53 & 324.89 \\ 4.46 & 2.53 & 39.94 & 1351.78 \\ 622.21 & 324.89 & 1351.78 & 208064.76 \end{bmatrix} \tag{2.22}$$

Chinstrap: $\eta_C = 0.194, \quad \boldsymbol{\mu}_C = (49.00,\ 18.48,\ 196.52,\ 3754.61)$

$$\Sigma_C = \begin{bmatrix} 9.97 & 2.10 & 7.12 & 531.50 \\ 2.10 & 1.21 & 4.00 & 240.67 \\ 7.12 & 4.00 & 48.09 & 1673.55 \\ 531.50 & 240.67 & 1673.55 & 144047.84 \end{bmatrix} \tag{2.23}$$

Gentoo: $\eta_G = 0.360, \quad \boldsymbol{\mu}_G = (47.50,\ 14.98,\ 217.19,\ 5076.02)$

$$\Sigma_G = \begin{bmatrix} 9.42 & 1.93 & 13.11 & 1031.17 \\ 1.93 & 0.95 & 4.46 & 352.80 \\ 13.11 & 4.46 & 41.71 & 2278.47 \\ 1031.17 & 352.80 & 2278.47 & 252067.11 \end{bmatrix} \tag{2.24}$$



Slika 2.2: Dijagram rasipanja po komponentama

Prikazan je dijagram rasipanja populacije pingvina po vrstama za vrijednosti numeričkih varijabli *duljina_kljuna, duljina_peraje* i *masa_tijela*. Možemo primijetiti da se razmještaji pingvina vrsta Adelie i Chinstrap preklapaju.

Algoritmu je trebalo 58 iteracija do postizanja konvergencije što vremenski odgovara trajanju od 5 minuta. Brzinu konvergencije računali smo formulom (2.20) te ona iznosi $r = 0.620$. Kada bi red veličine promatrane populacije bio veći $10^m, m \in \mathbb{N}$ puta, izvjesno

je da bi izvođenje algoritma trajalo znatno dulje. Sljedeći korak modeliranja populacije bila bi optimizacija računalnog algoritma. Smanjenje broja potrebnih iteracija možemo postići korištenjem neopadajućeg preslikavanja M u koraku maksimizacije. Budući da promatramo mješavinu triju višedimenzionalnih normalnih distribucija, trebalo je procijeniti $3K$ parametara. Spomenimo još Akaikeov i Bayesov informacijski kriterij koji penaliziraju modele s velikim brojem parametara, a oni za ovaj model iznose AIC = 10319.376 te BIC = 10353.889. Uz pretpostavku da populacija pingvina ima četiri ili više komponenata, računali smo pripadne informacijske kriterije i primijetili da se oni povećavaju kako broj komponenata raste. Time izvodimo zaključak da se optimalan model promatrane mješavine sastoji od tri komponente što i odgovara stvarnoj populaciji koju modeliramo. Populacija pingvina na kojoj smo implementirali algoritam pokazala se kao idealna za demonstraciju modeliranja Gaussovih mješavina.

Dodatak A

Računalni kôdovi u Pythonu

Ovdje su navedeni kôdovi korišteni za implementiranje EM algoritma za modeliranje populacije pingvina otočja Palmer. Za pisanje kôdova korišten je jezik *Python 3.7.4* i njegovi dodatni paketi *numpy* [10], *pandas* [6], *time, math, random, sys* [14], *scipy.stats* [16] i *matplotlib* [5].

A.1 Algoritam maksimizacije očekivanja

```
import numpy as np
import pandas as pd
import time
from time import sleep
import math
from scipy.stats import multivariate_normal as mvn
import linalg
import sys

start = time.time()

# --------------- FUNKCIJE ---------------#
def postignuta_konvergencija(eta_s, eta_n, mu_s, mu_n, sigma_s, sigma_n,
    delta_ograda, epsilon_ograda):
    zaustavljanje = 0
    log_vj_s = log_vjerodostojonost (eta_s, mu_s, sigma_s)
    log_vj_n = log_vjerodostojonost (eta_n, mu_n, sigma_n)
    if ( (( np.linalg.norm(np.asarray(eta_novi) - np.asarray(eta_stari)
    ,2) < delta_ograda ) and ( np.linalg.norm(np.asarray(mu_novi) - np.
    asarray(mu_stari),2)  < delta_ograda ) and ( np.linalg.norm(
    sigma_stari[0].as_matrix() - sigma_novi[0],  2) < delta_ograda ) and
     ( np.linalg.norm(sigma_stari[1].as_matrix() - sigma_novi[1],  2) <
    delta_ograda ) and ( np.linalg.norm(sigma_stari[2].as_matrix() -
```

```
    sigma_novi[2],  2) < delta_ograda ) ) or (np.abs(log_vj_n - log_vj_s
     ) < epsilon_ograda) ):
18          zaustavljanje = 1 #uvjet zaustavljanja je zadovoljen i
    postignuta je konvergencija
19
20      return(zaustavljanje)
21
22  def log_vjerodostojonost (trenutna_tezina, trenutno_ocekivanje,
    trenutna_kovarijacijska):
23      log_vj = 0
24      for i in range (N):
25          pomocna_suma = 0
26          vektor = znacajke.loc[i][:-1]
27          for k in range(K):
28              pomocna_suma += trenutna_tezina[k] * mvn.pdf(vektor, mean=
    trenutno_ocekivanje[k], cov=trenutna_kovarijacijska[k])
29          logaritam = np.log(pomocna_suma)
30          log_vj += logaritam
31      return(log_vj)
32
33  def brzina_konvergencije (eta_ps, eta_s, eta_n, mu_ps, mu_s, mu_n,
    sigma_ps, sigma_s, sigma_n):
34      brzina_konvergencije_eta = (np.linalg.norm(np.asarray(eta_n) - np.
    asarray(eta_s), 2))/(np.linalg.norm(np.asarray(eta_ps) - np.asarray(
    eta_s), 2))
35      brzina_konvergencije_mu = (np.linalg.norm(np.asarray(mu_n) - np.
    asarray(mu_s), 2))/(np.linalg.norm(np.asarray(mu_ps) - np.asarray(
    mu_s), 2))
36      brzina_konvergencije_sigma0 = (np.linalg.norm(np.matrix(sigma_n[0])
    - np.matrix(sigma_s[0]), 2))/(np.linalg.norm(np.matrix(sigma_ps[0])
    - np.matrix(sigma_s[0]), 2))
37      brzina_konvergencije_sigma1 = (np.linalg.norm(np.matrix(sigma_n[1])
    - np.matrix(sigma_s[1]), 2))/(np.linalg.norm(np.matrix(sigma_ps[1])
    - np.matrix(sigma_s[1]), 2))
38      brzina_konvergencije_sigma2 = (np.linalg.norm(np.matrix(sigma_n[2])
    - np.matrix(sigma_s[2]), 2))/(np.linalg.norm(np.matrix(sigma_ps[2])
    - np.matrix(sigma_s[2]), 2))
39      brzina = max(brzina_konvergencije_eta, brzina_konvergencije_mu,
    brzina_konvergencije_sigma0, brzina_konvergencije_sigma1,
    brzina_konvergencije_sigma2)
40      return(brzina)
41
42  def aic (broj_komponenti, log_v):
43      br_procijenjenih = 3*broj_komponenti
44      broj = 2* (br_procijenjenih-log_v)
45      return(broj)
46
```

```
def bic (broj_komponenti, broj_opazanja, log_v):
    n = broj_opazanja
    br_procijenjenih = 3*broj_komponenti
    k = br_procijenjenih
    broj = k * np.log(n) - 2 * (log_v)
    return(broj)
# --------------------------- FUNKCIJE ---------------------------#

podaci = pd.read_csv('potpuni_podaci.csv',header=0)
znacajke = podaci.copy()
z = znacajke.drop(['vrsta', 'otok'  , 'spol' ,  'godina'], axis=1)
znacajke = z

# ----------------------- INICIJALIZACIJA ----------------------- #
z0 = znacajke[znacajke["kMeans_pridruzivanje"].isin([0])]
znacajke_0 = z0.drop(['kMeans_pridruzivanje'], axis=1)

z1 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 1]
znacajke_1 = z1.drop(['kMeans_pridruzivanje'], axis=1)

z2 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 2]
znacajke_2 = z2.drop(['kMeans_pridruzivanje'], axis=1)

znacajke_num = znacajke.drop(['kMeans_pridruzivanje'], axis =1)

mu0 = znacajke_0.mean().astype(np.int)
mu1 = znacajke_1.mean().astype(np.int)
mu2 = znacajke_2.mean().astype(np.int)

mu_stari = (mu0, mu1, mu2)

sigma0 = znacajke_0.cov()
sigma1 = znacajke_1.cov()
sigma2 = znacajke_2.cov()

sigma_stari = (sigma0, sigma1, sigma2)

velicina_komponente0 = np.shape(znacajke_0)[0]
velicina_komponente1 = np.shape(znacajke_1)[0]
velicina_komponente2 = np.shape(znacajke_2)[0]

N = np.shape(znacajke)[0]
K = 3

eta0 = 1/N*velicina_komponente0
eta1 = 1/N*velicina_komponente1
eta2 = 1/N*velicina_komponente2
```

```
94
95 eta_stari = (eta0, eta1, eta2)
96
97
98 log_vjerodostojnost_stara = log_vjerodostojonost (eta_stari, mu_stari,
       sigma_stari)
99 # ---------------------- INICIJALIZACIJA ---------------------- #
100
101 # ------------------------ 1. ITERACIJA ------------------------ #
102 vjerojatnost = znacajke_num.copy()
103
104 vjerojatnost.rename(columns = {'duljina_kljuna':'komponenta0', '
       visina_kljuna' :'komponenta1', 'duljina_peraje':'komponenta2'},
       inplace=True)
105 vjerojatnost.drop(['masa_tijela'], inplace = True, axis=1)
106
107 for index, row in znacajke.iterrows():
108     i = index
109     vektor = znacajke.loc[i][:-1]
110     for j in range(3):
111         vjerojatnost.iloc[i,j] = mvn.pdf(vektor, mean=mu_stari[j], cov =
       sigma_stari[j])
112
113 odgovornost = vjerojatnost.copy()
114 odgovornost.rename(columns = {'komponenta0':0, 'komponenta1':1,'
       komponenta2':2}, inplace=True)
115 novo_pridruzivanje = odgovornost.idxmax(axis=1)
116
117 znacajke["kMeans_pridruzivanje"] = novo_pridruzivanje
118
119 z0 = znacajke[znacajke["kMeans_pridruzivanje"].isin([0])]
120 znacajke_0 = z0.drop(['kMeans_pridruzivanje'], axis=1)
121
122 z1 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 1]
123 znacajke_1 = z1.drop(['kMeans_pridruzivanje'], axis=1)
124
125 z2 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 2]
126 znacajke_2 = z2.drop(['kMeans_pridruzivanje'], axis=1)
127
128 znacajke_num = znacajke.drop(['kMeans_pridruzivanje'], axis =1)
129
130 for index, row in vjerojatnost.iterrows():
131     i = index
132     for j in range(K):
133         odgovornost.iloc[i,j] = vjerojatnost.iloc[i][j]*eta_stari[j]/np.
       dot(vjerojatnost.loc[i], eta_stari)
134
```

```
135 eta_novi = [1/N*odgovornost.sum(axis=0)[0], 1/N*odgovornost.sum(axis=0)
        [1], 1/N*odgovornost.sum(axis=0)[2]]
136
137 mu_novi = []
138 for k in range(K):
139     hy = 0
140     for i in range (N):
141         hy += odgovornost.iloc[i][k]*znacajke_num.loc[i]
142     hy = hy/odgovornost.sum(axis=0)[k]
143     mu_novi.append(hy)
144
145 sigma_n = []
146 for k in range (K):
147     sig = 0
148     for i in range(N):
149         sig += odgovornost.iloc[i][k] * np.outer(((znacajke_num.loc[i]-
        mu_novi[k]).to_numpy()), np.transpose((znacajke_num.loc[i]-mu_novi[k
        ]).to_numpy())) / odgovornost.sum(axis=0)[k]
150     sigma_n.append(sig)
151
152 sigma_novi = [pd.DataFrame(np.matrix(sigma_n[0])), pd.DataFrame(np.
        matrix(sigma_n[1])), pd.DataFrame(np.matrix(sigma_n[2]))]
153 log_vjerodostojnost_nova = log_vjerodostojonost (eta_novi, mu_novi,
        sigma_novi)
154
155 # ------------------------ 1. ITERACIJA ------------------------- #
156
157 br_iteracija = 1
158
159 delta = 1e-7
160 epsilon = 1e-5
161
162 if np.abs(log_vjerodostojnost_nova - log_vjerodostojnost_stara) < delta
        :
163     uvjet = 1
164 else:
165     uvjet = 0
166
167 # --------------------------- PETLJA ---------------------------- #
168
169 while uvjet == 0 :
170     br_iteracija += 1
171
172     trenutno_vrijeme = time.time()
173     elapsed = trenutno_vrijeme - start
174     minute = elapsed//60
175
```

```
176     sekunde = round(elapsed - minute*60)
177 
178     sys.stdout.write('\r')
179     sys.stdout.write("Vrijeme izvodjenja koda: [%d min : %d s]  ---
    Broj trenutne iteracije: %d"%(minute,sekunde, br_iteracija))
180 
181     sys.stdout.flush()
182     sleep(0.1)
183 
184     mu_prastari = mu_stari
185     eta_prastari = eta_stari
186     sigma_prastari = sigma_stari
187 
188     mu_stari = mu_novi.copy()
189     eta_stari = eta_novi.copy()
190     sigma_stari = sigma_novi.copy()
191 
192     log_vjerodostojnost_stara = log_vjerodostojnost_nova.copy()
193 
194     for index, row in znacajke.iterrows():
195         i = index
196         vektor = znacajke.loc[i][:-1]
197         for j in range(K):
198             vjerojatnost.iloc[i,j] = mvn.pdf(vektor, mean=mu_stari[j],
    cov=sigma_stari[j])
199 
200     odgovornost = vjerojatnost.copy()
201     odgovornost.rename(columns = {'komponenta0':0, 'komponenta1':1,'
    komponenta2':2}, inplace=True)
202 
203     novo_pridruzivanje = odgovornost.idxmax(axis=1)
204 
205     znacajke["kMeans_pridruzivanje"] = novo_pridruzivanje
206 
207     z0 = znacajke[znacajke["kMeans_pridruzivanje"].isin([0])]
208     znacajke_0 = z0.drop(['kMeans_pridruzivanje'], axis=1)
209 
210     z1 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 1]
211     znacajke_1 = z1.drop(['kMeans_pridruzivanje'], axis=1)
212 
213     z2 = znacajke.loc[znacajke['kMeans_pridruzivanje'] == 2]
214     znacajke_2 = z2.drop(['kMeans_pridruzivanje'], axis=1)
215 
216     znacajke_num = znacajke.drop(['kMeans_pridruzivanje'], axis =1)
217 
218     for index, row in vjerojatnost.iterrows():
219         i = index
```

```
220         for j in range(3):
221             odgovornost.iloc[i,j] = vjerojatnost.iloc[i][j]*eta_stari[j
    ]/np.dot(vjerojatnost.loc[i], eta_stari)
222
223
224     eta_novi = [1/N*odgovornost.sum(axis=0)[0], 1/N*odgovornost.sum(axis
    =0)[1], 1/N*odgovornost.sum(axis=0)[2]]
225
226     mu_novi = []
227     for k in range(K):
228         hy = 0
229         for i in range (N):
230             hy += odgovornost.iloc[i][k]*znacajke_num.loc[i]
231         hy = hy/odgovornost.sum(axis=0)[k]
232         mu_novi.append(hy)
233
234     sigma_n = []
235     for k in range (K):
236         sig = 0
237         for i in range(N):
238             sig += odgovornost.iloc[i][k] * np.outer(((znacajke_num.loc[
    i]-mu_novi[k]).to_numpy()), np.transpose((znacajke_num.loc[i]-
    mu_novi[k]).to_numpy())) / odgovornost.sum(axis=0)[k]
239         sigma_n.append(sig)
240
241     sigma_novi = [pd.DataFrame(np.matrix(sigma_n[0])), pd.DataFrame(np.
    matrix(sigma_n[1])), pd.DataFrame(np.matrix(sigma_n[2]))]
242
243     log_vjerodostojnost_nova = log_vjerodostojonost (eta_novi, mu_novi,
    sigma_novi)
244
245     je_li_iskonvergiralo = postignuta_konvergencija(eta_stari, eta_novi,
     mu_stari, mu_novi, sigma_stari, sigma_novi, delta, epsilon)
246     uvjet = je_li_iskonvergiralo
247
248
249 brzina = brzina_konvergencije(eta_prastari, eta_stari, eta_novi,
    mu_prastari, mu_stari, mu_novi, sigma_prastari, sigma_stari,
    sigma_novi)
250 print()
251 print()
252 print("Brzina konvergencije je ", brzina)
253 print()
254
255
256 iznos_a = aic (K, log_vjerodostojnost_nova)
257 iznos_b = bic (K, N, log_vjerodostojnost_nova)
```

```
258
259 print()
260 print("Akaikeov informacijski kriterij iznosi  AIC =  ", iznos_a)
261 print()
262 print()
263 print("Bayesov informacijski kriterij iznosi   BIC =  ", iznos_b)
264 print()
265 # -------------------------- PETLJA -------------------------- #
266
267 # ----------------------- KRAJ ALGORITMA ----------------------- #
268
269 with open(r'sredine.txt', 'w') as f1:
270     f1.write("ocekivanje\n")
271     for item in mu_novi:
272         f1.write("%s\n" % item)
273
274 with open(r'kovarijacijske.txt', 'w') as f2:
275     for item in sigma_novi:
276         f2.write("%s\n" % item)
277
278 with open(r'tezine.txt', 'w') as f3:
279     f3.write("tezina\n")
280     for item in eta_novi:
281         f3.write("%s\n" % item)
282
283 print("Nakon ",br_iteracija, ". iteracije, zavrsavamo s algoritmom i
    analiziramo rezultate.")
284
285 podaci['EM_pridruzivanje'] = novo_pridruzivanje
286 podaci.to_csv(r'konacna_tablica.csv', header=True, index=False)
287
288 trenutno_vrijeme = time.time()
289 end = time.time()
290 elapsed = end - start
291 minute = elapsed//60
292 sekunde = round(elapsed - minute*60)
293
294 print ("Ukupno vrijeme za postizanje konvergencije pod ovim uvjetima je
    ", minute,"minuta i ",sekunde, "sekundi.")
295 print("vektor tezina:", eta_novi)
296 print("vektor ocekivanja:", mu_novi)
297 print("kovarijacijske matrice:", sigma_novi)
298
299 print("------------------------ KRAJ ------------------------")
```

A.2 Algoritam k–srednjih vrijednosti ++

```
import numpy as np
import pandas as pd
import time
import random
import math

penguins_data = pd.read_csv("penguins.csv")
penguins = penguins_data.drop(columns=['Unnamed: 0'])
podaci = penguins.rename(columns = {'species':'vrsta', 'island':'otok',
    'bill_length_mm':'duljina_kljuna', 'bill_depth_mm':'visina_kljuna',
    'flipper_length_mm':'duljina_peraje' ,'body_mass_g':'masa_tijela','
    sex':'spol', 'year':'godina'})
potpuni_podaci = podaci.dropna(subset=['duljina_kljuna', 'visina_kljuna'
    , 'duljina_peraje' , 'masa_tijela'])
opazanja = potpuni_podaci.reset_index()[['duljina_kljuna', '
    visina_kljuna', 'duljina_peraje' , 'masa_tijela']].values.tolist()

br_tocaka = len(opazanja)
tocke = np.array(opazanja)
dimenzija_tocaka = np.size(tocke[0])
K = 3
br_centara = K

def euklidska_metrika(x,y):
    dist = 0
    for i in range(len(x)):
        dist = dist + (x[i]-y[i])*(x[i]-y[i])
    return dist

# inicijalizacija centroida
centri = []
centri.append(tocke[random.randint(0,br_tocaka-1)][:])
centri

for i in range(br_centara-1):
    maxi = 0
    pom = 0;

    for j in range(br_tocaka):
        for z in range(np.size(centri, axis=0)):
            if euklidska_metrika(tocke[j][:],centri[z]) > maxi:
                maxi = euklidska_metrika(tocke[j][:],centri[z])
                pom = j
```

```
    centri.append(tocke[pom][:])

group_id = []
for i in range(br_tocaka):
    group_id.append(0)

br_iteracija = 100

for temp in range(br_iteracija):
    # korak pridruzivanja
    for i in range(br_tocaka):
        udaljenost_trenutna = euklidska_metrika(tocke[i], centri[0])
        k = 0

        for j in range(br_centara):
            udaljenost_pom = euklidska_metrika(tocke[i], centri[j])
            if udaljenost_pom < udaljenost_trenutna:
                udaljenost_trenutna = udaljenost_pom
                k = j
        group_id[i] = k

    # korak racunanja novih centroida
    for i in range(br_centara):
        niz = []
        for j in range(dimenzija_tocaka):
            niz.append(0)
        cluster_count = 0
        for j in range(br_tocaka):
            if group_id[j] == i:
                cluster_count += 1
                niz += tocke[j]

        centil95 = np.percentile(centri[i], 95)
        if cluster_count > 0:
            for j in range(dimenzija_tocaka):
                if centri[i][j] > centil95:
                    centri[i][j] = niz[j]*1.0/cluster_count

indeksi = list(potpuni_podaci.index.values)
pridruzivanja = pd.Series(group_id, index = indeksi)
potpuni_podaci['kMeans_pridruzivanje'] = pridruzivanja
potpuni_podaci.to_csv(r'potpuni_podaci.csv',header=True, index=False)
```

Bibliografija

[1] A. Dempster, N. Laird i D. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society. Series B (Methodological) **39** (1977), br. 1, 1–38.

[2] J. Demšar et al., Orange: Data Mining Toolbox in Python, Journal of Machine Learning Research **14** (2013), 2349–2353, `http://jmlr.org/papers/v14/demsar13a.html`, version 3.23.1.

[3] W. Feller, On a General Class of "Contagious" Distributions, Annals of Mathematical Statistics **14** (1943), br. 4, 389–400.

[4] S. Frühwirth-Schnatter, Finite Mixture and Markov Switching Models, Springer Science + Business Media, LLC, New York, 2006.

[5] J. D. Hunter, Matplotlib: A 2D graphics environment, Computing in science & engineering **9** (2007), br. 3, 90–95.

[6] W. McKinney et al., Data structures for statistical computing in python, Proceedings of the 9th Python in Science Conference, sv. 445, Austin, TX, 2010, str. 51–56.

[7] G. J. McLachlan, S. X. Lee i S. I. Rathnayake, Finite Mixture Models, Annual Review of Statistics and Its Application **6** (2019), br. 1, 355–378.

[8] G. J. McLachlan i D. Peel, Finite Mixture Models, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc, New York, 2000.

[9] S. K. Ng, T. Krishnan i G. J. Mclachlan, The EM algorithm, Papers/Humboldt–Universität Berlin, Center for Applied Statistics and Economics (CASE) **24** (2004).

[10] T. E. Oliphant, A guide to NumPy, **1**, Trelgol Publishing USA, 2006.

[11] K. Pearson, Contributions to the Mathematical Theory of Evolution, Philosophical Transactions of the Royal Society of London **A, 185** (1894), 71–110.

[12] P. E. Rossi, G. M. Allenby i R. McCulloch, Bayesian Statistics and Marketing, Wiley Series in Probability and Statistics, John Wiley & Sons, Ltd, West Sussex, 2005.

[13] N. Sarapa, Teorija vjerojatnosti, Školska knjiga, Zagreb, 2002.

[14] G. Van Rossum, The Python Library Reference, release 3.8.2, Python Software Foundation, 2020.

[15] G. Van Rossum i F. L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009, version 3.7.4.

[16] P. Virtanen et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, 2020.

[17] C. F. J. Wu, On the Convergence Properties of the EM Algorithm, The Annals of Statistics **11** (1983), br. 1, 95–103.

[18] J. Šnajder i B. Dalbelo Bašić, Strojno učenje, 2014, `https://www.fer.unizg.hr/_download/repository/StrojnoUcenje.pdf`, datum pristupa (11.10.2019.).

Sažetak

Model Gaussovih mješavina često se pojavljuje u raznim znanstvenim istraživanjima. Mješavina je populacija koja se sastoji od komponenata čije vjerojatnosne distribucije mogu, ali ne moraju nužno, pripadati istoj familiji distribucija. U ovom smo radu proučavali Gaussove mješavine čiji je broj komponenata konačan i unaprijed poznat. Za takve mješavine konstruirali smo model tako što smo procijenili distribucijske parametre svake komponente pa komponentama pridružili opažanja. Koristili smo metode procjene parametara iz teorije frekvencionističke i Bayesovske statistike koje su na kraju i objedinjene algoritmom maksimizacije očekivanja (*EM algoritam*). Svaki model mješavine potrebno je prilagoditi dostupnim informacijama o populaciji. Najjednostavniji slučaj koji modeliramo je kada su poznate gustoće komponenata pa im još preostaje pridružiti jedinke korištenjem Bayesovog naivnog klasifikatora. Ako su gustoće komponenata nepoznate, ali je poznata realizacija pridruživanja svakog opažanja, potrebno je samo procijeniti parametre gustoća. U tom smo slučaju procjenu proveli metodom maksimalne vjerodostojnosti kao i Bayesovom metodom. Za zadnji slučaj smo ostavili primjer populacije za koju unaprijed ne znamo niti parametre gustoća komponenata niti realizaciju pridruživanja jedinki. Taj je problem jednostavno riješen pomoću EM algoritma koji je u radu teorijski objašnjen, ali i realiziran na stvarnoj populaciji. Proces modeliranja Gaussovih mješavina ilustrirali smo na populaciji tri vrste pingvina i pritom za svaku jedinku odredili pridruživanje pomoću EM algoritma. Model smo izrazili kao linearnu kombinaciju višedimenzionalnih normalnih gustoća čije smo parametre procijenili.

Summary

Gaussian mixture model occurs commonly in various scientific research. A mixture is a population that consists of components whose probability distributions could, but should not necessarily, belong to the same family of distributions. This thesis studies Gaussian mixtures whose number of components is finite and known in advance. For those mixtures, we constructed a model in a way that the distribution parameters were estimated and observations were assigned to the components. Parameter estimation methods that were used come both from frequentist and Bayesian statistics and are consolidated with the Expectation–Maximization algorithm (*the EM algorithm*). Each mixture model should be adjusted according to the available information on the population. The simplest case that is modelled is the one where densities of the components are known and only allocation of the observation remains. Allocation is conducted by using the naïve Bayes classifier. In the case where component densities are unknown and the allocation of the observations is known, only parameter estimation is required. Unknown parameters are estimated by using the method of maximum likelihood estimation along with the Bayesian method of estimation. The last studied case is the example of a population for which both the component parameters and the allocation is unknown. That problem is easily solved by using the EM algorithm which is in this thesis theoretically explained and implemented on the real–life population. The process of modelling Gaussian mixtures is illustrated on the population of three penguin species for which all the units are allocated to the components by using the EM algorithm. The model is given as a linear combination of multivariate normal distributions whose parameters were estimated.

Životopis

Rođena sam 14. kolovoza 1994. u Zagrebu. Srednju školu, zagrebačku XV. gimnaziju, upisujem 2009. godine, a preddiplomski studij Matematika na Prirodoslovno matematičkom fakultetu – Matematičkom odsjeku 2013. godine. Diplomski studij Matematička statistika upisujem na istom fakultetu 2018. godine.