UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Ana Nikolić

BAJESOV METOD POTPORNIH VEKTORA

master rad

Beograd, 2021.

Mentor:

dr Bojana Milošević, docent
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Marko Obradović, docent
Univerzitet u Beogradu, Matematički fakultet

dr Aleksandar Kartelj, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Naslov master rada: Bajesov metod potpornih vektora

Rezime: U ovom radu opisana je reprezentacija metoda potpornih vektora, koja pomoću skrivenih promenljivih omogućava upotrebu alata često korišćenih u Bajesovoj statistici, kao što su ocenjivanje parametara EM ili MCMC algoritmima. Pokazano je da je minimizacija funkcije gubitka metoda potpornih vektora ekvivalentna pronalaženju mode pseudo aposteriorne raspodele nepoznatih parametara modela. Korišćenjem skrivenih promenljivih, pseudo aposteriornu raspodelu moguće je predstaviti kao mešavinu normalnih raspodela. U radu je opisana implementacija algoritma i predstavljeni su rezultati primene modela nad različitim skupovima podataka.

Ključne reči: Metod potpornih vektora, Bajesova statistika, EM, MCMC

Sadržaj

Glava 1

Uvod

Metod potpornih vektora jedan je od najpopularnijih metoda binarne klasifikacije. Razvijao ga je Vladimir Vapnik sa svojim kolegama iz kompanije *AT&T Bell Laboratories*. Model je pokazao izuzetne rezultate u brojnim problemima mašinskog učenja, zbog čega je i stekao svoju popularnost. Jedna od primena na kojoj su radili istraživači iz $AT\&T$-a je prepoznavanje rukopisa (v. [2]).

Model je neprobabilistički i primenom nad novim podacima ne daje direktno informaciju o verovatnoći pripadnosti određenim klasama. Takođe, za rešavanje optimizacionog problema, tačnije za minimizaciju funkcije gubitka, koriste se specijalno konstruisani numerički algoritmi.

Navedeni potencijalni nedostatci mogu biti rešeni Bajesovom reprezentacijom ovog modela, koja je 2011. godine opisana u radu Nikolasa Polsona i Stivena Skota (v. [19]). Ovim pristupom, minimizacija funkcije gubitka metoda potpornih vektora postaje ekvivalentna nalaženju mode pseudo aposteriorne raspodele parametara modela. Ovu raspodelu moguće je predstaviti kao mešavinu normalnih raspodela uvođenjem skrivenih promenljivih, koje se dalje koriste za konstrukciju EM ili MCMC algoritama za ocenjivanje parametara modela.

1.1 Organizacija rada

U glavi 2 predstavljen je metod potpornih vektora. Opisana je upotreba regularizacije u ovom metodu i izvedena je funkcija gubitka. U glavi 3 opisan je metod maksimalne verodostojnosti, kao i Bajesov metod ocene nepoznatih parametara. Opisan je i način rada EM i MCMC algoritma u opštem slučaju. U glavi 4 dokazane

su glavne teoreme o reprezentaciji aposteriorne raspodele nepoznatih parametara mešavinom normalnih raspodela. Izvedene su i uslovne raspodele skrivenih promenljivih, koje su dalje primenjene na konstrukciju EM i MCMC algoritma za konkretan slučaj ovog modela. Ponašanje ovog modela na realnim podacima i uporedni prikaz rezultata za različite nivoe regularizacije, dat je u glavi 5. Glava 6 sumira utiske o prednostima i manama ovog pristupa. Na kraju, implementacija modela pisana je u programskom jeziku *R* i kodovi najvažnijih funkcija priloženi su u glavi 7.

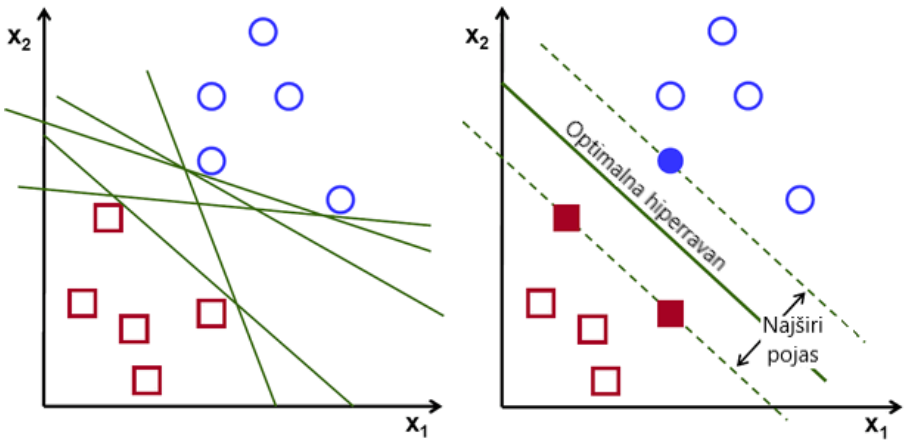
Glava 2

Metod potpornih vektora

Metod potpornih vektora (eng. *support vector machine*) jedan je od popularnijih modela nadgledanog učenja. Osnovna verzija ovog metoda rešava problem binarne klasifikacije. Neka imamo skup za obučavanje D koji čini n tačaka.

$$D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^k, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

Ideja je naći hiperravan, takvu da su sve tačke jedne klase sa iste strane te hiperravni, a tačke različitih klasa sa suprotnih strana. Ukoliko takva hiperravan postoji, dati skup tačaka nazivamo linearno razdvojivim. Bez dodatnih uslova takva hiperravan ne mora biti jedinstvena, ali nisu sve podjednako dobre. Prema ovom metodu, optimalna hiperravan je ona sa najvećom razdaljinom do najbliže tačke skupa za obučavanje. Drugim rečima, ovaj metod maksimizuje širinu pojasa oko razdvajajuće hiperravni, zbog čega se i naziva metodom zasnovanim na širokom pojasu (eng. *large margin*).



Slika 2.1: Izbor optimalne hiperravni

2.1 Metod potpornih vektora sa tvrdim pojasom

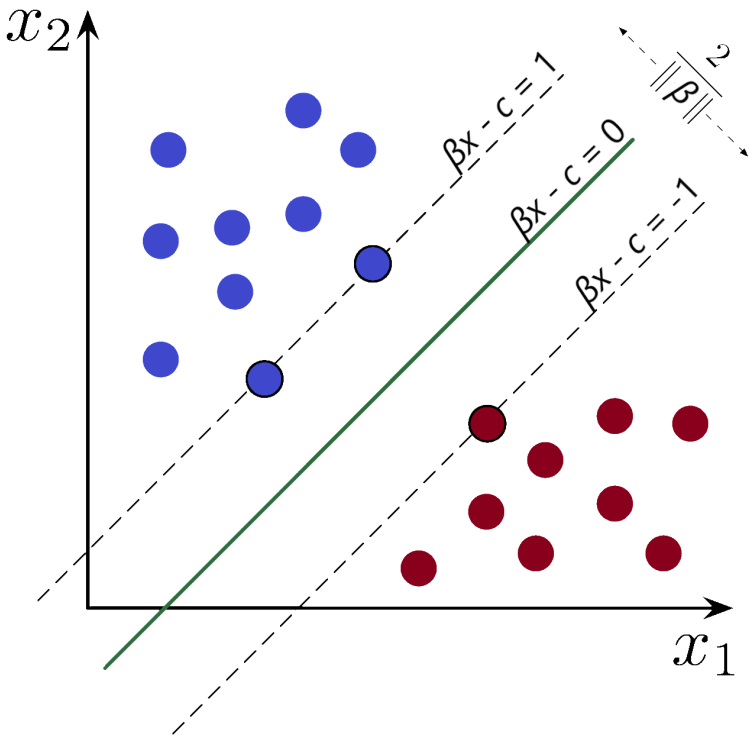
Metod potpornih vektora sa tvrdim pojasom pretpostavlja linearnu razdvojivost klasa. U suprotnom, tražena hiperravan ne postoji.

Jednačina proizvoljne hiperravni je $\boldsymbol{\beta x} - \mathbf{c} = 0$, gde je $\boldsymbol{\beta}$ vektor normale na tu hiperravan. Ako je uslov linearne razdvojivosti zadovoljen, onda možemo odabrati dve paralelne hiperravni koje razdvajaju klase, takve da je rastojanje između njih najveće moguće. Optimalna hiperravan tada naleže tačno u sredini između ove dve, tj. ona je podjednako udaljena od najbližih tačaka obe klase. Jednačine hiperravni paralelnih optimalnoj su $\boldsymbol{\beta x} - \mathbf{c} = a$ i $\boldsymbol{\beta x} - \mathbf{c} = -a$. Deljenjem ovih jednačina sa a i preimenovanjem koeficijenata $\boldsymbol{\beta}/a$ i $\mathbf{c}/a$ nazad u $\boldsymbol{\beta}$ i $\mathbf{c}$ imamo:

$$\boldsymbol{\beta x} - \mathbf{c} = 1$$

$$\boldsymbol{\beta x} - \mathbf{c} = -1$$

Tačke najbliže optimalnoj hiperravni, tj. tačke koje pripadaju hiperravnima paralelnim optimalnoj, nazivaju se potpornim vektorima.



Slika 2.2: Hiperravni paralelne optimalnoj leže na potpornim vektorima

Rastojanje tačke od hiperravni dato je sa:

$$\frac{|\boldsymbol{\beta x} - \mathbf{c}|}{\|\boldsymbol{\beta}\|_2}$$

Ukoliko je ta tačka potporni vektor, važi $|\boldsymbol{\beta x} - \mathbf{c}| = 1$, pa je rastojanje koje treba maksimizovati, tj. rastojanje između najbližih pripadnika različitih klasa $\frac{2}{\|\boldsymbol{\beta}\|_2}$. Uz to je potrebno da važi i da su sve tačke sa odgovarajuće strane optimalne hiperravni, kao i da su na većem rastojanju od nje u odnosu na potporne vektore, koji su na rastojanju 1. $\forall(\mathbf{x}_i, y_i) \in D$:

$$\boldsymbol{\beta x_i} - \mathbf{c} \geq 1, y_i = 1$$

$$\boldsymbol{\beta x_i} - \mathbf{c} \leq -1, y_i = -1$$

Ekvivalentno, ovaj optimizacioni problem se može napisati kao:

$$\min_{\beta} \frac{\|\boldsymbol{\beta}\|_2}{2}$$

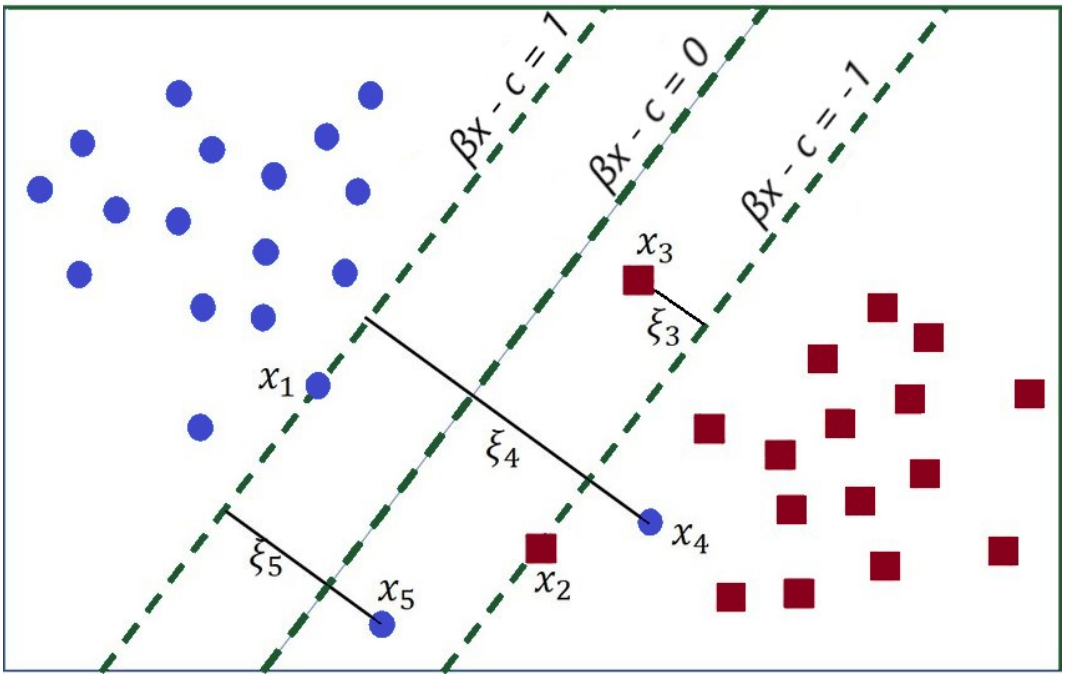
$$y_i(\boldsymbol{\beta x_i} - \mathbf{c}) \geq 1, \qquad i = 1, ..., N$$

Rešavanjem ovog optimizacionog problema dobija se optimalna hiperravan u odnosu na koju se određuju klase bilo koje nove tačke $\mathbf{x}$, kao $y = sgn(\boldsymbol{\beta x} - \mathbf{c})$.

2.2 Metod potpornih vektora sa mekim pojasom

Pretpostavka o linearnoj razdvojivosti klasa je prejaka da bi ovakav metod u praksi našao stvarnu primenu. Metod potpornih vektora sa mekim pojasom dozvoljava preklapanje klasa, tj. prihvata postojanje tačaka koje će se naći sa pogrešne strane hiperravni određenih potpornim vektorima. Ideja o pronalaženju najšireg pojasa oko optimalne hiperravni ostaje ista, ali se toleriše određen broj grešaka, tj. tačaka koje kad ne bi postojale, pojas oko optimalne hiperravni bi u potpunosti razdvajao klase. Potrebno je da ovakvih grešaka bude što manje. Zbog toga se uvode nove pormenljive $\xi_i, i = 1, ..., N$, koje mere udaljenost tačaka od hiperravni određenih potpornim vektorima odgovarajućih klasa, ali samo u slučaju da se one nalaze sa pogrešne strane. Za tačke $\mathbf{x}_i$, koje su sa prave strane, $\xi_i = 0$.

Kao i u prethodnom slučaju, potrebno je naći $\min_\beta \frac{\|\boldsymbol{\beta}\|_2}{2}$, ali je sada uz ovo potrebno i da greške ξ_i budu što manje, kao i da su sve tačke sa odgovarajućih



Slika 2.3: Metod potpornih vektora sa mekim pojasom

strana hiperravni, ili ako nisu, da su na rastojanju ξ_i od njih. Poslednji uslov može se zapisati kao:

$$y_i(\boldsymbol{\beta x_i} - \mathbf{c}) \geq 1 - \xi_i, \qquad i = 1, ..., N$$

Optimizacioni problem se tada svodi na:

$$\min_{\beta} \frac{\|\boldsymbol{\beta}\|_2}{2} + C \sum_{i=1}^{N} \xi_i$$

$$y_i(\boldsymbol{\beta x_i} - \mathbf{c}) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \qquad i = 1, ..., N$$

U slučaju da je promenljiva $\xi_i > 0$, tada važi $y_i(\boldsymbol{\beta x_i} - \mathbf{c}) < 1$, a ξ_i je onda baš jednako $\xi_i = 1 - y_i(\boldsymbol{\beta x_i} - \mathbf{c})$. Suprotno, $\xi_i = 0$ odgovara onim tačkama za koje važi $y_i(\boldsymbol{\beta x_i} - \mathbf{c}) > 1$. Iz ovoga sledi da je

$$\xi_i = max(0, 1 - y_i(\boldsymbol{\beta x_i} - \mathbf{c})),$$

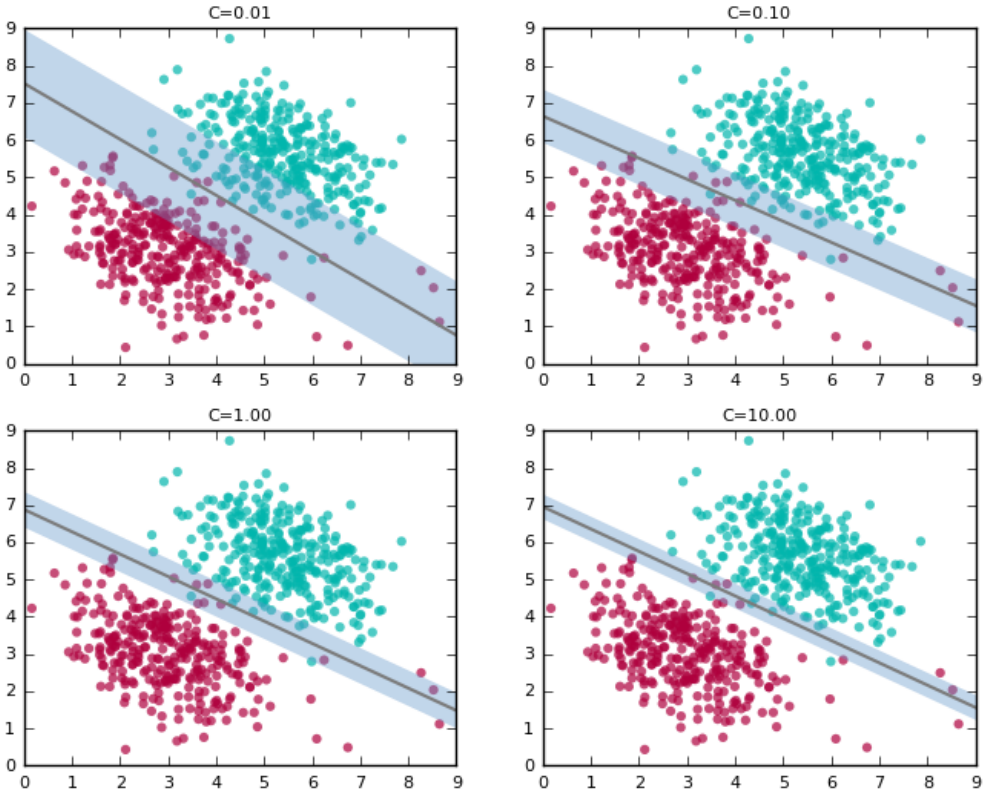
a prethodni problem može se preformulisati u

$$\min_{\beta} \frac{\|\boldsymbol{\beta}\|_2}{2} + C \sum_{i=1}^{N} max(0, 1 - y_i(\boldsymbol{\beta x_i} - \mathbf{c})).$$

U slučaju metoda potpornih vektora sa mekim pojasom, potpornim vektorima se pored onih koji se nalaze na ivici pojasa, nazivaju i sve tačke skupa za obučavanje

koje se nalaze unutar pojasa, kao i one koje se nalaze sa suprotne strane pojasa u odnosu na ostale predstavnike svoje klase. Primetimo da model zavisi samo od ovih tačaka. One tačke koje se nalaze izvan pojasa nikako ne utiču na rezultat optimizacionog problema, dokle god ostanu sa ispravne strane pojasa.

Kod ovog metoda, regularizacija je uključena u samoj formulaciji optimizacionog problema. Metaparametar C određuje koliko se značaja prilikom optimizacije pridodaje greškama, tj. koliko njih je prihvatljivo da postoji u modelu. Metaparametar C zapravo kontroliše nagodbu između pristrasnosti i disperzije modela. Ukoliko je C veliko, greške modela su izuzetno važne pa parametri ξ_i teže da budu 0, što rezultuje uskim pojasom. U tom slučaju je potpornih vektora malo, pa dobijeni model može imati nisku pristrasnost, ali visoku vrednost disperzije. Suprotno, ako je C malo, to dovodi do jače regularizacije. Tolerancija prema greškama je veća, pa je i pojas širi. Potpornih vektora ima više i ovakvi modeli mogu imati nešto višu pristrasnost, ali nižu disperziju. Čak, ukoliko bi bilo $C = 0$, greške modela bi mogle biti proizvoljno velike, jer nisu uopšte važne. Optimalno rešenje takvog problema je $\boldsymbol{\beta} = 0$.



Slika 2.4: Širina pojasa u odnosu na parametar C

Optimizacioni problem se može ekvivalentno zapisati i kao

$$\sum_{i=1}^{N} max(0, 1 - y_i(\boldsymbol{\beta x_i} - \mathbf{c})) + \lambda \|\boldsymbol{\beta}\|_2. \tag{2.1}$$

Regularizacioni izraz stoji uz parametar λ i ima veću težinu za veće vrednosti λ, suprotno parametru C. Funkcija greške je oblika $L(u, v) = max(0, 1 - uv)$ i naziva se funkcija greške u vidu šarke (eng. *hinge loss*).

Rešenje ovog problema dobija se numerički. U te svrhe su konstruisani razni algoritmi, jedan od kojih je SMO algoritam (eng. *Sequential minimal optimization*), koji je 1998. godine razvio Džon Plat (v. [18]).

Rešenje je oblika

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} \alpha_i y_i x_i,$$

gde su α_i Lagražovi množioci za koje važi $0 \leq \alpha_i \leq C$. Za x_i koji odgovaraju potpornim vektorima važi da je $\alpha_i > 0$. Za sve tačke koje nisu potporni vektori važi $\alpha_i = 0$, samim tim rešenje uopšte ne zavisi od njih (v. [13]).

Model je dat funkcijom

$$f_{\hat{\beta}}(\boldsymbol{x}) = \hat{\boldsymbol{\beta}}\boldsymbol{x} - \mathbf{c} = \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot \boldsymbol{x}) - \mathbf{c},$$

a klasa nove instance se određuje kao znak date funkcije

$$sgn(f_{\hat{\beta}}(\boldsymbol{x})).$$

Glava 3

Ocena parametara

Kao sastavni deo statističkog modela, često su uključeni razni parametri koje je potrebno oceniti kako bi model bio u potpunosti određen. Iz skupa mogućih parametara, treba odabrati one, pri kojima model najbolje opisuje podatke nad kojima je treniran. Tačnije, pri dobro odabranim parametrima modela, verovatnoća realizovanog uzorka bi trebalo da bude što viša. Jedan od metoda ocenjivanja parametara koji je zasnovan na ovom principu je metod maksimalne verodostojnosti.

3.1 Metod maksimalne verodostojnosti

Neka su $\mathbf{X} = (X_1, ..., X_n)$ slučajne veličine sa zajedničkom gustinom raspodele $f(\boldsymbol{x}; \theta)$, gde su $\theta \in \Theta$ svi nepoznati parametri koji određuju raspodelu od $\mathbf{X}$. Na primer, u slučaju da važi $\mathbf{X} \sim N(\mu, \sigma^2)$, tada je $\theta = (\mu, \sigma^2)$ iz skupa Θ, gde je $\Theta = \{(\mu, \sigma^2) | \mu \in \mathbb{R}, \sigma^2 > 0\}$.

Funkcija verodostojnosti se definiše kao:

$$L(\theta; \mathbf{x}) = f(\mathbf{x}; \theta) = \prod_{i=1}^{n} f\left(x_i; \theta\right)$$

Funkcija verodostojnosti je zajednička raspodela realizovanog uzorka $(x_1, ..., x_n)$ i zavisi od nepoznatog parametra θ. U prethodnom izrazu, druga jednakost važi samo ukoliko su komponente uzorka, X_i, $i = 1, ..., n$ nezavisne i jednako raspodeljene.

Definicija 3.1.0.1 *Neka je za uzorak $\boldsymbol{x} = (x_1, ..., x_n)$, funkcija $L(\theta)$ najveća po Θ, za $\theta = S(\boldsymbol{x})$, odnosno*

$$\sup_{\theta \in \Theta} L(\theta) = L(S(\boldsymbol{x})).$$

Tada je $\hat{\theta} = S(\boldsymbol{X})$ *ocena maksimalne verodostojnosti parametra* θ*.*

Dakle ocena metodom maksimalne verodostojnosti je ona vrednost parametra za koju je funkcija verodostojnosti najveća. Takva ocena, ukoliko postoji, ne mora biti jedinstvena.

U praksi se češće umesto maksimuma funkcije verodostojnosti traži maksimum logaritma te funkcije, zbog pogodnijih analitičkih osobina:

$$l(\theta) = \log L(\theta, \mathbf{x}) = \sum_{i=1}^{n} \log f\left(x_i, \theta\right).$$

3.2 Bajesov metod

Parametri koji učestvuju u modelu se obično smatraju nepoznatim konstantama čija vrednost se može oceniti pomoću dostupnih podataka. Nasuprot tome, Bajesov metod nepoznati parametar θ opisuje funkcijom raspodele nad prostorom Θ. Pre posmatranja podataka, parametru θ je dodeljena *apriorna* raspodela sa gustinom $\pi(\theta)$. Pomoću nje se izražava nivo uverenja o vrednostima, iz prostora Θ, koje parametar θ može imati. Drugim rečima, iskazuje se subjektivna pretpostavka o mogućim vrednostima parametra θ. Zajednička gustina $f(\mathbf{x}; \theta)$ se smatra uslovnom gustinom, pod uslovom datog parametra θ. Uzimajući u obzir podatke, apriorna raspodela se ažurira uz pomoć Bajesove teoreme i ovako nastala raspodela $\theta|\mathbf{X}$ naziva se *aposteriorna*.

Gustina aposteriorne raspodele parametra θ uz dato $\mathbf{X} = \boldsymbol{x}$ se definiše kao:

$$\pi(\theta \mid \boldsymbol{x}) = \frac{f(\boldsymbol{x}; \theta)\pi(\theta)}{\int_{\Theta} f(\boldsymbol{x}; t)\pi(t)dt}.$$

Primetimo da u prethodnoj formuli, izraz u imeniocu ne zavisi od parametra θ, pa sledi

$$\pi(\theta \mid \boldsymbol{x}) \propto f(\boldsymbol{x}; \theta)\pi(\theta) = L(\theta)\pi(\theta),$$

gde je $L(\theta)$ funkcija verodostojnosti.

Ovim pristupom, parametar θ je opisan aposteriornom funkcjiom raspodele nakon opažanja datih podataka, uzevši u obzir i apriornu raspodelu. Sva informacija o nesigurnosti u odabiru parametra θ opisana je aposteriornom raspodelom.

Pretpostavimo da imamo parametar θ čija je apriorna raspodela $\theta \sim \Gamma(\alpha, \beta)$. Ova raspodela takođe zavisi od nekih parametara α i β. Ovi parametri se često zovu *hiperparametri*. Kako je prema Bajesovom metodu apriornu raspodelu neophodno odrediti nezavisno od podataka koji su dostupni, ove parametre je takođe potrebno odrediti *a priori*. Međutim, taj korak je moguće izbeći ako se ovi parametri ocene iz podataka, a onda se tako dobijene vrednosti koriste u apriornoj raspodeli. Neka je $\pi(\theta; \gamma)$ gustina apriorne raspodele koja zavisi od hiperparametra γ. U ovom primeri bi bilo $\gamma = (\alpha, \beta)$. Tada možemo definisati *marginalnu* zajedničku gustinu kao:

$$g(\boldsymbol{x}; \gamma) = \int_{\Theta} f(\boldsymbol{x}; \theta)\pi(\theta; \gamma)d\theta$$

Hiperparametar γ se onda može oceniti, na primer, metodom maksimalne verodostojnosti, koristeći zajedničku gustinu $g(\boldsymbol{x}; \gamma)$ kao funkciju verodostojnosti. Ovaj metod iako dosta koristan, zapravo suštinski nije u potpunosti Bajesov, jer se prilikom određivanja apriorne raspodele nepoznatog parametra koriste podaci. Iz tog razloga ovaj pristup pripada takozvanom empirijskom Bajesovom zaključivanju.

Ocena parametra θ može se dobiti kao neka deskriptivna statistika aposteriorne raspodele. Često se koristi srednja vrednost, medijana ili moda, tačka u kojoj aposteriorna raspodela dostiže najveću vrednost.

Iako se apriorna raspodela može odabrati proizvoljno, zgodno je odabrati takvu raspodelu da je iz nje aposteriornu raspodelu lako moguće izvesti.

Definicija 3.2.0.1 *Neka je $F_\theta = \{f(\boldsymbol{x}; \theta) : \theta \in \Theta\}$ familija gustina koje zavise od parametra θ. Kažemo da je familija apriornih raspodela P konjugovana familija od F, ako za svako $f_\theta \in F$ važi da aposteriorna raspodela parametra θ pripada P.*

Odabirom takve apriorne raspodele, aposteriornu je moguće direktno izvesti bez upotrebe numeričkih metoda. Jedan od primera su raspodele koje pripadaju eksponencijalnoj familiji raspodela.

Definicija 3.2.0.2 *Neka je $\mathbf{X} \sim F_\theta, \theta = (\theta_1, ..., \theta_p)$. Kažemo da je F_θ k-parametarska eksponencijalna familija raspodela ako se $f_\theta \in F_\theta$ može predstaviti kao:*

$$f(\boldsymbol{x}; \theta) = \exp\Big(\sum_{i=1}^{k} c_i(\theta)T_i(x) - d(\theta) + S(x))\Big),$$

pri čemu nosač raspodele $supp(f(\boldsymbol{x}; \theta))$ ne zavisi od parametra θ.

Primetimo da u prethodnoj definiciji k ne mora biti jednako p. Primeri raspodela koje pripadaju eksponencijalnoj familiji su normalna raspodela, gama, beta, bernulijeva, eksponencijalna raspodela, itd.

Primer 3.2.0.1 *Familija gama raspodela $\gamma(a,b)$, $a,b>0$ je dvoparametarska eksponencijalna familija raspodela.*

$\mathbf{X}=(X_1,...,X_n)$, $\mathbf{X}\sim\gamma(a,b)$

$$f(x\mid a,b)=\frac{b^a}{\Gamma(a)}x^{a-1}\exp(-bx),\qquad x>0$$

$$f(x\mid a,b)=\prod_{i=1}^{n}f(x_i\mid a,b)=\exp\Big((a-1)\sum_{i=1}^{n}lnx_i-b\sum_{i=1}^{n}x_i+n\cdot a\cdot lnb-n\cdot ln\Gamma(a)\Big)$$

Ovde je $c_1(\theta)=a-1$, $T_1(\theta)=\sum_{i=1}^{n}lnx_i$, $c_2(\theta)=-b$, $T_2(\theta)=\sum_{i=1}^{n}x_i$.

Primer 3.2.0.2 *Familija gama raspodela $\gamma(a,b)$, $a,b>0$ je konjugovana familija od $\varepsilon(\theta)$.*

$$\begin{aligned}\pi(\theta\mid x)&\propto f(x;\theta)\pi(\theta)\\&\propto\prod_{i=1}^{n}\theta e^{-\theta x_i}\frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta}\\&\propto\theta^{a+n-1}e^{-(b+n\overline{x})\theta}\end{aligned}$$

odakle je $\theta\mid x\sim\gamma(a+n,b+n\overline{x})$.

Aposteriorna raspodela takođe pruža osnovu u predviđanju vrednosti novih opservacija. Neka je $\tilde{x}$ nova opservacija. Tada je:

$$\mathrm{P}\left(\tilde{x}\mid\mathbf{X}\right)=\int f\left(\tilde{x}\mid\theta\right)\cdot\pi(\theta\mid\mathbf{X})d\theta. \tag{3.1}$$

Ovaj pristup kao rezultat daje raspodelu verovatnoća koja uzima u obzir nesigurnost prilikom ocenjivanja nepoznatog parametra. Nasuprot tome, drugim metodama, kao na primer metodom maksimalne verodostojnosti, prvo se pronađe optimalna vrednost nepoznatog parametra $\hat{\theta}$, a onda se pomoću gustine rapsposele $f(\tilde{x}\mid\hat{\theta})$, i fiksne vrednosti ocenjenog parametra, dobije raspodela koja zanemaruje moguću grešku ocenjivanja, čime se varijacija prediktivne raspodele potcenjuje.

Konjugovane familije apriornih raspodela obično se koriste radi lakšeg izračunavanja aposteriorne raspodele. Iako one imaju lepa svojstva, ne mora biti slučaj da se njima subjektivna uverennja o nepoznatom parametru mogu dovoljno dobro opisati. Porastom performansi modernih računara, jednostavnost izračunavanja postaje manje važna i sve više se koriste numerički metodi ocenjivanja parametara.

3.3 EM algoritam

Prilikom ocenjivanja metodom maksimalne verodostojnosti, često nije moguće doći do konkretnog izgraza za traženu ocenu. U tim slučajevima ona se određuje numerički. Jedan od mogućih metoda je *algoritam očekivanja i maksimizacije*, u daljem teksu EM algoritam.

EM algoritam je jedan od najpoznatijih metoda pronalaženja ocena metodom maksimalne verodostojnosti i ima široku primenu u statistici. Iako se pojavljivao i u ranijim radovima, naziv kakav ima danas dobio je 1977. u radu v. [3].

Najčešće se koristi u slučajevima kada postoje nedostajući podaci, ali se takođe može primeniti i u slučaju da u modelu postoje određene *skrivene* (eng. *latent*) promenljive $\mathbf{z}$. U tom slučaju, skrivene promenljive se tretiraju isto kao da su nedostajući podaci. One takođe nisu opažene, ali za razliku od nedostajućih podataka, njih nije ni predviđeno opažati. Uz pretpostavku da su skriveni parametri empirijski poznati, njihova uloga je da olakšaju ocenjivanje parametara, onda kada je ocena parametara raspodele $f(\boldsymbol{x};\theta)$ teška. Obzirom da parametri modela nisu poznati, prvi korak EM algoritma je određivanje inicijalnih vrednosti parametara. Nakon toga, algoritam čine dva koraka:

- E-korak: Izračunavanje očekivane vrednosti logaritma verodostojnosti $\mathbb{E}[l(\theta;\mathbf{x},\mathbf{z})]$
- M-korak: Ocenjivanje parametara tako da očekivanje iz prethodnog koraka bude najveće.

EM algoritam u svakom koraku poboljšava ocenu iz prethodne iteracije. Međutim, algoritam zavisi od odabira početnih vrednosti nepoznatih parametara, pa je moguće da iskonvergira ka lokalnom maksimumu umesto ka globalnom. Iz ovog razloga se prepotučuje da se algoritam pokrene više puta, za različite vrednosti inicijalnih parametara.

Neka su $\mathbf{X}$ dostupni podaci, $\mathbf{Z}$ skrivene promenljive i θ parametri modela. Želimo da maksimizujemo funkciju verodostojnosti $f(\mathbf{x};\theta)$ po θ. Za proizvoljnu raspodelu $q(\mathbf{z})$ imamo:

$$\begin{aligned}\log f(\mathbf{x} \mid \theta) &= \log f(\mathbf{x} \mid \theta) \int q(\mathbf{z}) d\mathbf{z} \\ &= \int q(\mathbf{z}) \log \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{f(\mathbf{z} \mid \mathbf{x}, \theta)} d\mathbf{z} \\ &= \int q(\mathbf{z}) \log \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{q(\mathbf{z})} d\mathbf{z} - \int q(\mathbf{z}) \log \frac{f(\mathbf{z} \mid \mathbf{x}, \theta)}{q(\mathbf{z})} d\mathbf{z} \\ &= \mathcal{L}(q, \theta) + KL(q \| f),\end{aligned}$$

gde su

$$\begin{aligned}\mathcal{L}(q, \theta) &= \int q(\mathbf{z}) \log \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{q(\mathbf{z})} d\mathbf{z} \\ KL(q \| f) &= -\int q(\mathbf{z}) \log \frac{f(\mathbf{z} \mid \mathbf{x}, \theta)}{q(\mathbf{z})} d\mathbf{z}.\end{aligned}$$

Sa KL označavamo Kulbak-Lajblerovo razilaženje između raspodela $q(\mathbf{z})$ i $f(\mathbf{z} \mid \mathbf{x}, \theta)$, koje predstavlja meru sličnosti između dve raspodele. Važi da je $KL \geq 0$, odakle sledi:

$$\log f(\mathbf{x} \mid \theta) \geq \mathcal{L}(q, \theta).$$

S' obzirom na to da $f(\mathbf{x} \mid \theta)$, zbog skrivenih promenljivih, ne može direktno da se maksimizuje, želimo da maksimizujemo $\mathcal{L}(q, \theta)$. $\mathcal{L}(q, \theta)$ se maksimizuje iterativnim procesom. Tokom E-koraka, parametar θ je fiksiran, nazovimo ga θ^{t-1}, a $q(\mathbf{z})$ biramo tako da maksimituje $\mathcal{L}(q, \theta^{t-1})$. Najveće $\mathcal{L}(q, \theta^{t-1})$ je ono za koje je $KL(q \| f) = 0$, a to se dešava samo u slučaju $q = f$, tj.

$$q(\mathbf{z}) = f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right).$$

Dalje, tokom M-koraka, $q(\mathbf{z})$ je fiksirano, a θ se bira tako da maksimizuje $\mathcal{L}(q, \theta)$.

$$\begin{aligned}\mathcal{L}(q, \theta) &= \int f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right) \log \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right)} d\mathbf{z} \\ &= \int f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right) \log f(\mathbf{x}, \mathbf{z} \mid \theta) d\mathbf{z} - \int f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right)\right) \log f\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right) d\mathbf{z}.\end{aligned}$$

Desni deo prethodnog izraza ne zavisi od θ, pa je dovoljno maksimizovati samo levi deo. Označimo ga sa

$$Q\left(\theta, \theta^{t-1}\right) = \int p\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right) \log f(\mathbf{x}, \mathbf{z} \mid \theta) d\mathbf{z}.$$

Dakle, EM algoritam ima sledeće korake:

Algoritam:

1. Odabir inicijalne vrednosti parametra θ^{t-1}
2. E-korak: Odrediti očekivanje

$$\begin{aligned}\mathbb{E}[l(\theta;\mathbf{x},\mathbf{z})] &= \int p\left(\mathbf{z} \mid \mathbf{x}, \theta^{t-1}\right) \log f(\mathbf{x},\mathbf{z} \mid \theta) d\mathbf{z} \\ &= Q\left(\theta, \theta^{t-1}\right)\end{aligned}$$

3. M-korak: Odrediti θ^t tako da važi

$$\theta^t = \arg\max_{\theta} Q\left(\theta, \theta^{t-1}\right).$$

4.

$$\theta^{t-1} \leftarrow \theta^t$$

Koraci 2 - 4 se ponavljaju do konvergencije.

3.4 MCMC algoritam

Kod velikog broja probabilističkih metoda, direktno izvođenje zaključaka analitički često nije moguće i neophodno je osloniti se na numeričke aproksimacije. Do sada je metod maksimalne verodostojnosti bio izabrani metod ocenjivanja parametara. Ukoliko želimo da se oslonimo na Bajesov metod i pronalaženje aposteriorne raspodele nepoznatih parametara, da bi izvođenje zaključaka o njihovim vrednostima bilo moguće, neophodno je uzorkovanje iz rezultujuće aposteriorne raspodele. To može biti računski veoma zahtevno. Jedna popularna klasa algoritama za uzorkovanje iz aposteriorne raspodele je Monte Karlo algoritam sa lancima Markova (na dalje MCMC). Osnovna ideja je da se konstruiše lanac Markova koji je ergodičan i stacionaran, čija granična raspodela je upravo ona iz koje želimo da generišemo uzorak. Iako elementi lanca Markova nisu međusobno nezavisne slučajne veličine, kao što je slučaj kod uzoraka generisanim standardnim Monte Karlo metodama, uzorak dobijen MCMC metodom zaista jeste iz tražene raspodele (v. [1]).

Definicija 3.4.0.1 *Neka je* $\{X_n, n \in \mathbb{N}\}$ *slučajni proces sa prebrojivim skupom stanja* S*. Kažemo da je dati proces lanac Markova ako ispunjava Markovljevo svojstvo: Za svako* $n \in \mathbb{N}$ *i za sve* $i_1, ..., i_n \in S$ *važi:*

$$P\{X_n = i_n \mid X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\} = P\{X_n = i_n \mid X_{n-1} = i_{n-1}\}$$

Markovljevo svojstvo suštinski govori da stanje u kome će se proces naći u budućem trenutku zavisi samo od trenutnog stanja procesa. Dodatno, lanac je homogen ako

$$(\forall i, j \in \mathbb{Z}) \qquad p_{i,j} = P\{X_n = j \mid X_{n-1} = i\}$$

ne zavisi od izbora n. U suprotnom je lanac nehomogen. Verovatnoće $p_{i,j}$ nazivamo verovatnoćama prelaska iz stanja i u stanje j. Matrica verovatnoće prelaska je matrica $\mathbf{P} = [p_{i,j}]_{i,j\in\mathbb{Z}}$ i za nju važi da je zbir elemenata po svim redovima jednak jedinici.

Definicija 3.4.0.2 *Neka je $\{X_n, n \in \mathbb{N}_0\}$ homogeni lanac Markova sa skupom stanja $\mathbb{Z}$. Kažemo da je stanje $j \in \mathbb{Z}$ <u>dostižno</u> iz stanja $i \in \mathbb{Z}$ ako je*

$$\sum_{i=1}^{\infty} P\{X_1 \neq j, X_2 \neq j, ..., X_{n-1} \neq j, X_n = j \mid X_0 = 1\} > 0$$

.

Stanja i i j međusobno komuniciraju ako je stanje j dostižno iz stanja i i obratno.

Definicija 3.4.0.3 *Skup S je <u>nerazloživ</u> ukoliko proizvoljna dva stanja iz tog skupa međusobno komuniciraju.*

Definicija 3.4.0.4 *Kažemo da je Markovljev lanac <u>nesvodljiv</u> ukoliko mu je ceo skup stanja nerazloživ.*

Definicija 3.4.0.5 *Neka je $\{X_n, n \in \mathbb{N}_0\}$ homogeni lanac Markova sa skupom stanja $\mathbb{Z}$. Raspodela verovatnoća $(\pi_i)_{i\in\mathbb{Z}}, \sum_i = 1$ je <u>stacionarna raspodela</u> tog lanca ako važi:*

$$(\forall i \in \mathbb{Z})(\forall n \in \mathbb{N}_0) \qquad P\{X_0 = i\} = \pi_i \Rightarrow P\{X_n = i\} = \pi_i$$

Ekvivalentan iskaz sa prethodnim je sledeći:

$\pi = (\pi_i)_{i\in\mathbb{Z}}$ je stacionarna raspodela u odnosu na matricu $\mathbf{P}$ *akko* važi

$$(\forall n \in \mathbb{N})\pi = \pi\mathbf{P}^n$$

Definicija 3.4.0.6 *Markovljev lanac je <u>ergodičan</u> ako važi:*

$$(\forall i, j \in \mathbb{Z})\exists \lim_{n\to\infty} p_{ij}(n) = p_j > 0$$

pri čemu p_j ne zavisi od i i važi

$$\sum_{j\in\mathbb{Z}} p_j = 1$$

Raspodela $(p_j)_{j\in\mathbb{Z}}$ je granična raspodela.

Veza između ergodičnosti lanca i stacionarne raspodele je sledeća. Naime, nesvodljivi lanac Markova ima stancionarnu raspodelu akko je ergodičan. Takođe, ako je lanac ergodičan, onda ima jedinstvenu stacionarnu raspodelu, koja je upravo jednaka graničnoj raspodeli.

Primetimo i sledeće: Ako je π stacionarna raspodela homogenog lanca Markova sa skupom stanja S, a $\mathbf{P}$ njegova matrica prelaska, tada važi

$$\int_S \mathbf{P}(x,y)\pi(x)dx = \pi(y)$$

Cilj je iskoristiti svojstva lanaca Markova da bi se mogao izvući uzorak iz tražene raspodele. To je moguće ukoliko uzmemo ergodičan lanac Markova čija je granična raspodela baš ona koja nam je potrebna. Uz to važi da će ovakav lanac uvek iskonvergirati željenoj raspodeli bez obzira na izbor početne tačke.

Gibsov algoritam

Gibsov algoritam je primer MCMC algoritma, koji je specijalan slučaj Metropolis - Hejstings algoritma (v. [20]). Neka je $p(\theta) = p(\theta_1, ..., \theta_k)$ raspodela iz koje je potrebno izvući uzorak i neka je $\theta^{(\mathbf{0})}$ početno stanje lanca Markova. Svaki korak Gibsovog algoritma podrazumeva izvlačenje uzorka jedne slučajne veličine, čija je raspodela uslovljena trenutnim vrednostima svih ostalih slučajnih veličina. Ovaj postupak je primenljiv kada tražena zajednička raspodela nije eksplicitno poznata ili je teško uzorkovati iz nje direktno, ali je uslovna raspodela svake promenljive poznata ili je iz nje lakše uzorkovati. Ovakav niz uzoraka čini Markovljev lanac, a može se pokazati da je granična raspodela upravo tražena zajednička raspodela.

Pretpostavimo da želimo da dobijemo N uzoraka iz zajedničke raspodele slučajnog vektora $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$.

Algoritam:

1. Inicijalizacija lanca: $\theta^{(\mathbf{0})} = \left(\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}\right)$.

2. Za $i = 0, ..., N-1$:
   i za $j = 1, ..., k$:

$$\theta_j^{(i+1)} \sim p\left(\theta_j \mid \theta_1^{(i+1)}, ..., \theta_{j-1}^{(i+1)}, \theta_{j+1}^{(i)}, \ldots, \theta_k^{(i)}\right).$$

Dakle, za svaku od N iteracija (za svaki od N elemenata uzorka), svakoj slučajnoj veličini iz zajedničke raspodele dodeljuje se vrednost koja je izvučena is raspodele uslovljene vrednostima svih ostalih slučajnih veličina iz te iteracije, ili prve prethodne ukoliko u tekućoj iteraciji toj slučajnoj veličini još nije dodeljena vrednost.

Treba dokazati da se datim algoritmom zaista dobija uzorak iz tražene raspodele. Pre svega, raspodela $p(\theta^{(i)})$ mora biti ista za svaki i, tj. raspodela mora biti stacionarna. Ovo važi jer prilikom uzorkovanja iz raspodele $p(\theta_i \mid \theta \backslash i)$, marginalna raspodela $p(\theta \backslash i)$ ne zavisi od θ_i, pa je u svakom koraku ista. Kako uslovna i marginalna raspodela određuju zajedničku raspodelu, ona je takođe ista za svako i, tj. raspodela je stacionarna.

$$p(\theta_j \mid \theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_k) = \frac{p(\theta_1, \ldots, \theta_k)}{p(\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \theta_k)} \propto p(\theta_1, \ldots, \theta_k)$$

Uslov ergodičnosti takođe mora biti zadovoljen. Dovoljan uslov ergodičnosti lanca je da nijedna od uslovnih raspodela nije nigde nula. U tom slučaju je iz svake tačke, iz skupa stanja lanca Markova, moguće doći do bilo koje duge tačke nakon konačno mnogo koraka, tj. lanac je nesvodljiv. Kako je granična raspodela stacionarna, on je onda i ergodičan. Ukoliko ovaj uslov nije zadovoljlen, ergodičnost se mora eksplicitno dokazati.

Da bi algoritam bio potpuno određen, neophodno je definisati i raspodelu inicijalnog stanja lanca, iako će nakon dovoljno mnogo iteracija, raspodele elemenata lanca postati praktično nezavisne od početne raspodele. Zbog toga što su susedni elementi lanca Markova međusobno visoko korelisani, ukoliko je potrebno doći do približno nezavisnih slučajnih veličina, lanac je neophodno „proreditiˮ, tj. potrebno je uzvući određeni poduzorak, koji , na primer, može biti sačinjen od svakog m-tog elementa lanca.

Pošto uzorci iz početnih iteracija nisu bliski onima iz aposteriorne raspodele, praktikuje se primena *zagrevanja*, tj. određeni broj uzoraka na početku se ignoriše. Nakon što lanac dostigne stacionarnost, marginalnu raspodelu bilo kog podskupa

slučajnih veličina iz zajedničke raspodele je moguće odrediti iz dobijenog uzorka. Ili ukoliko je eksplicina forma uslovnih raspodela, koje učestvuju u algoritmu, poznata, tada se raspodela slučajne veličine, npr. X_k može dobiti na sledeći način:

$$\widehat{\mathrm{P}}_{X_k}(u) = \frac{1}{(N-n)} \sum_{t=n+1}^{N} \mathrm{P}\left(u \mid X_\ell^{(t)}, \ell \neq k\right). \tag{3.2}$$

U prethodnom izrazu se prvih n uzoraka odbacuje, nakon kojih se proces stabilizuje i dostiže se stacionarnost (v. [7]).

Glava 4

Bajesov metod potpornih vektora

Metod potpornih vektora za klasifikaciju, na osnovu prediktora $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_k)$, $\mathbf{X}_j \in \mathbb{R}^n$, $j = 1, ..., k$, zavisnoj promenljivoj y_i, $i = 1, ..., n$ dodeljuje vrednost iz skupa $\{-1, 1\}$. Optimizacioni izraz koji je potrebno minimizovati, sa L^α, $\alpha \in (0, 2]$ regularizacijom, je:

$$d_\alpha(\beta, \lambda) = \sum_{i=1}^{n} \max\left(1 - y_i \mathbf{x}_i^T \beta, 0\right) + \lambda \sum_{j=1}^{k} |\beta_j|^\alpha , \tag{4.1}$$

gde je λ parametar regularizacije, a $\mathbf{x}_i^T \beta$ je skalarni proizvod i-te realizacije prediktora i parametra $\beta = (\beta_1, ..., \beta_k)$. Pretpostavka je da su podaci nad kojima se model obučava standardizovani, tj. da je standardna devijacija svakog prediktora jednaka jedan. Kao što je pomenuto u glavi 2 na strani 8, ovaj optimizacioni problem se rešava numerički.

Minimizacija izraza 4.1 ekvivalentna je pronalaženju mode raspodele $p(\beta \mid \lambda, \alpha, y)$ definisane na sledeći način:

$$\begin{aligned} p(\beta \mid \lambda, \alpha, y) &= C_\alpha(\lambda) L(y \mid \beta) p(\beta \mid \lambda, \alpha) \\ &\propto \exp\left(-d_\alpha(\beta, \lambda)\right). \end{aligned} \tag{4.2}$$

$C_\alpha(\lambda)$ predstavlja konstantu normalizacije koja osigurava da prethodnim izrazom raspodela dobro definisana. $p(\beta \mid \lambda, \alpha)$ je apriorna raspodela parametra β, takva da važi

$$p(\beta \mid \lambda, \alpha) \propto \exp\left(-2\lambda \sum_{j=1}^{k} |\beta_j|^\alpha\right),$$

a $L(y \mid \beta)$ je aproksimacija funkcije verodostojnosti koja nije normalizovana po y, zbog čega se zove i pseudo funkcija verodostojnosti:

$$L(y \mid \beta) = \prod_i L_i(y_i \mid \beta) = \exp\left\{-2\sum_{i=1}^{n} \max\left(1 - y_i \mathbf{x}_i^T \beta, 0\right)\right\}. \tag{4.3}$$

U prethodnom izrazu, svako $L_i(y_i \mid \beta) = \exp\left\{-2\max\left(1 - y_i \mathbf{x}_i^T \beta, 0\right)\right\}$ moguće je zameniti normalizacijom $\tilde{L}_i(y_i \mid \beta) = L_i(y_i \mid \beta)/\big(L_i(y_i \mid \beta) + L_i(-y_i \mid \beta)\big)$, čime bi L postala prava funkcija verodostojnosti, jer y_i može biti samo 1 ili -1. Međutim, korišćenjem funkcije L kao u 4.3, dobija se model koji odgovara standardnom metodu potpornih vektora. Zbog korišćenja pseudo funkcije verodostojnosti, raspodela $p(\beta \mid \lambda, \alpha, y)$, definisana u 4.2, naziva se pseudo aposteriorna raspodela (v. [19]).

4.1 Reprezentacija mešavinom normalnih raspodela

Uvođenjem određenih skrivenih promenljivih, metod potpornih vektora je moguće interpretirati bajesovski. Tačnije, pomoću njih je pseudo funkciju verodostojnosti moguće predstaviti kao mešavinu normalnih raspodela, što dalje omogućava ocenjivanje parametara modela tehnikama kao što su MCMC ili EM algoritam. Skrivene promenljive $\{\nu_i\}_{i=1,\ldots,n}$ se definišu tako da važi da je $L_i(y_i \mid \beta)$ marginalna raspodela zajedničke raspodele $L_i(y_i, \nu_i \mid \beta)$.

Teorema 4.1.0.1 *Pseudo funkcija verodostojnosti za svako y_i može se izraziti kao:*

$$\begin{aligned} L_i(y_i \mid \beta) &= \exp\left\{-2\max\left(1 - y_i \mathbf{x}_i^T \beta, 0\right)\right\} \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\nu_i}} \exp\left(-\frac{1}{2}\frac{\left(1 + \nu_i - y_i \mathbf{x}_i^T \beta\right)^2}{\nu_i}\right) d\nu_i. \end{aligned} \tag{4.4}$$

Dokaz: Da bi dokazali ovu teoremu potrebno je prvo dokazati da važi sledeći identitet:

$$\int_0^\infty \phi(u \mid -\nu, \nu) d\nu = e^{-2\max(u,0)}. \tag{4.5}$$

Kao što je pokazano u radu [8], za svako $a, b > 0$ važi:

$$\int_0^\infty \frac{a}{\sqrt{2\pi\nu}} e^{-\frac{1}{2}\left(a^2\nu + b^2\nu^{-1}\right)} d\nu = e^{-|ab|}.$$

Zamenom $a = 1$, i $b =$ i množenjem prethodnoh izraza sa e^{-u} dobijamo:

$$\int_0^\infty \frac{1}{\sqrt{2\pi\nu}} e^{-\frac{u^2}{2\nu} - u - \frac{1}{2}\nu} d\nu = e^{-|u|-u}.$$

Korišćenjem poznatog identiteta $2max(u, 0) = |u| + u$ dobijamo traženi identitet

$$\int_0^\infty \frac{1}{\sqrt{2\pi\nu}} e^{-\frac{(u+\nu)^2}{2\nu}} d\nu = e^{-2\max(u,0)}.$$

Dokaz teoreme sada sledi iz prethodnog izraza zamenom ν sa ν_i i $u = 1 - y_i \mathbf{x}_i^T \beta$. □

Odgovarajući rezultat se može izvesti i za apriornu raspodelu parametra β. Ona se može zapisati u obliku generalizovane normalne raspodele (v. [14]) čija je funkcija gustine:

$$p(\beta \mid \lambda, \alpha) = \prod_{j=1}^{k} p\left(\beta_j \mid \lambda, \alpha\right) = \left(\frac{\alpha\lambda}{2\Gamma\left(\alpha^{-1}\right)}\right)^k \exp\left(-\sum_{i=1}^{k} |\lambda\beta_j|^\alpha\right). \tag{4.6}$$

Specijalni slučajevi za $\alpha = 2$ ili $\alpha = 1$ odgovaraju gustinama normalne i Laplasove raspodele. Ti slučajevi su ujedno i najvažniji jer se za $\alpha = 1$ radi o $LASSO$ regularizaciji (v. [23]), a za $\alpha = 2$ o grebenoj regresiji (eng. *ridge regression*) (v. [5]).

Definicija 4.1.0.1 *Neka su X_1 i X_2 nezavisne kopije slučajne veličine X. Kažemo da je X stabilna ako za svako $a, b > 0$, $aX_1 + bX_2$ ima istu raspodelu kao $cX + d$ za neke $c > 0$ i $d \in \mathbb{R}$.*

Funkciju gustine stabilne raspodele, u opštem slučaju, nije moguće predstaviti analitički, ali njenu karakterističnu funkciju jeste. Specijalno, slučajna veličina sa pozitivnom stabilnom raspodelom indeksa α ima karakterističnu funkciju oblika

$$\varphi(t) = \exp\left(-\gamma|t|^\alpha[1 + i\,\mathrm{sgn}(t)\tan(\alpha\pi/2)]\right),$$

gde je $0 < \alpha < 1$, $\gamma > 0$ i nosač raspodele je $[0, \infty)$ (v. [17]) .

Apriornu raspodelu je takođe moguće izraziti kao mešavinu normalnih raspodela. Raspodele iz familije generalizovanih normalnih raspodela, tačnije raspodele čija je gustina oblika:

$$f(x) = Ce^{-|x|^\alpha}, \qquad x \in \mathbb{R},$$

mogu se predstaviti kao mešavina normalnih raspodela. To je dokazano za slučaj $\alpha \in [1, 2]$ u radu [25], a kasnije je, u radu [6], prošireno na $\alpha \in (0, 1]$. Opšti slučaj je dat sledećom teoremom.

Teorema 4.1.0.2 *Apriorna raspodela parametra β može se izraziti kao mešavina normalnih raspodela*

$$p\left(\beta_j \mid \lambda, \alpha\right) = \int_0^\infty \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right) p\left(\omega_j \mid \alpha\right) d\omega_j, \tag{4.7}$$

gde je $p\left(\omega_j \mid \alpha\right) \propto \omega_j^{-\frac{3}{2}} St_{\frac{\alpha}{2}}^{+}\left(\omega_j^{-1}\right)$, a $St_{\frac{\alpha}{2}}^{+}$ je funkcija gustine pozitivne stabilne slučajne promenljive sa indeksom $\alpha/2$. Specijano, za $\alpha = 1$ važi

$$p\left(\beta_j \mid \lambda, \alpha = 1\right) = \int_0^\infty \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right) \frac{1}{2} e^{-\frac{\omega_j}{2}} d\omega_j. \tag{4.8}$$

Na ovaj način je određen još jedan skup skrivenih promenljivih $\{\omega_i\}_{i=1,\ldots,k}$.

Iz prethodne dve teoreme sledi da za zajedničku pseudo aposteriornu raspodelu parametra β i skrivenih promenljivih $\{\nu_i\}_{i=1,\ldots,n}$ i $\{\omega_i\}_{i=1,\ldots,k}$ važi:

$$\begin{aligned} p(\beta, \nu, \omega \mid y, \lambda, \alpha) \propto & \prod_{i=1}^{n} \nu_i^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n} \frac{\left(1 + \nu_i - y_i \mathbf{x}_i^T \beta\right)^2}{\nu_i}\right) \\ & \times \prod_{j=1}^{k} \omega_j^{-\frac{1}{2}} \exp\left(-\frac{\lambda^2}{2}\sum_{j=1}^{k} \frac{\beta_j^2}{\omega_j}\right) p\left(\omega_j \mid \alpha\right). \end{aligned} \tag{4.9}$$

Konačno, pseudo aposteriorna raspodela parametra β, dobija se kao marginalna raspodela prethodne po skrivenim parametrima ν i ω,

$$p(\beta \mid \lambda, \alpha, y) = \int p(\beta, \nu, \omega \mid \lambda, \alpha, y) d\nu d\omega.$$

Uz ovako definisanu raspodelu nepoznatog parametra β, algoritmi za ocenjivanje parametara kao što su EM algoritam ili MCMC mogu biti primenjeni.

4.2 Određivanje uslovnih raspodela

Neka je $\Omega = diag(\omega)$, $N = diag(\nu)$, a $\mathbf{X}$ matrica dimenzije $n \times k$ čiji je i-ti red jednak $y_i \mathbf{x_i}$.

Da bi se odredila uslovna raspodela parametra parametra β, model se, kao u radu [19], može zapisati u hijerarhijskom obliku

$$\begin{aligned} \mathbf{1} + \nu &= \mathbf{X}\beta + N^{\frac{1}{2}} \epsilon^{\nu} \\ \beta &= \frac{1}{\lambda} \Omega^{\frac{1}{2}} \epsilon^{\beta}, \end{aligned}$$

gde su ϵ^ν i ϵ^β vektori nezavisnih slučajnih veličina sa standardnom normalnom raspodelom, čuje dimenzije odgovaraju dimenzijama parametara ν i β.

Sledi da je aposteriorna raspodela parametra β takođe normalna

$$p(\beta \mid \lambda, \nu, \omega, y) \sim \mathcal{N}(b, B),$$

gde su parametri raspodele

$$B^{-1} = \lambda^2 \Omega^{-1} + \mathbf{X}^T N^{-1} \mathbf{X} \text{ i } b = B\mathbf{X}^T \left(\mathbf{1} + \nu^{-1}\right).$$

Definicija 4.2.0.1 *Slučajna veličina X ima inverznu Gausovu raspodelu $X \sim \mathcal{IG}(\mu, \lambda)$, sa parametrima μ i λ, ako je njena funkcija gustine*

$$p(x \mid \mu, \lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right),$$

gde je $x \in (0, \infty)$ i parametri $\mu, \lambda > 0$.

Za slučajnu veličinu sa inverznom Gausovom raspodelom važi da je $EX = \mu$, a $DX = \mu^3/\lambda$ (v. [4]).

Definicija 4.2.0.2 *Slučajna veličina X ima generalizovanu inverznu Gausovu raspodelu $X \sim \mathcal{GIG}(\gamma, \psi, \chi)$, sa parametrima γ, ψ i χ, ako je njena funkcija gustine*

$$p(x \mid \gamma, \psi, \chi) = C(\gamma, \psi, \chi) x^{\gamma-1} \exp\left(-\frac{1}{2}\left(\frac{\chi}{x} + \psi x\right)\right),$$

gde je $C(\gamma, \psi, \chi)$ odgovarajuća konstanta normalizacije, $x \in (0, \infty)$ i parametri $\gamma \in \mathbb{R}$ i $\psi, \chi > 0$.

Važi sledeća osobina (v. [9]):
Ako slučajna veličina X ima generalizovanu inverznu Gausovu raspodelu $X \sim \mathcal{GIG}(1/2, \lambda, \lambda/\mu^2)$, tada X^{-1} ima inverznu Gausovu raspodelu $X \sim \mathcal{IG}(\mu, \lambda)$.

Korišćenjem ove osobine, može se dokazati posledica teoreme 4.1.0.1.

Posledica 4.2.0.1 *Uslovna raspodela skrivenog parametra ν_i je*

$$p\left(\nu_i^{-1} \mid \beta, y_i\right) \sim \mathcal{IG}\left(\left|1 - y_i \mathbf{x}_i^T \beta\right|^{-1}, 1\right).$$

Dokaz: Iz teoreme 4.1.0.1 sledi

$$\begin{aligned} p\left(\nu_i \mid \beta, y_i\right) &\propto \frac{1}{\sqrt{2\pi\nu_i}} \exp\left\{-\frac{\left(1-y_i\mathbf{x}_i^T\beta-\nu_i\right)^2}{2\nu_i}\right\} \\ &\propto \frac{1}{\sqrt{2\pi\nu_i}} \exp\left\{-\frac{1}{2}\left(\frac{\left(1-y_i\mathbf{x}_i^T\beta\right)^2}{\nu_i}+\nu_i\right)\right\} \\ &\sim \mathcal{GIG}\left\{\frac{1}{2}, 1, \left(1-y_i\mathbf{x}_i^T\beta\right)^2\right\}. \end{aligned}$$

Iz prethodne osobine sledi

$$p\left(\nu_i^{-1} \mid \beta, y_i\right) \sim \mathcal{IG}\left(\left|1-y_i\mathbf{x}_i^T\beta\right|^{-1}, 1\right),$$

što je trebalo dokazati. □

Analogno prethodnom, uslovna raspodela skrivenog parametra ω_j može se izvesti iz izraza 4.7 u teoremi 4.1.0.2. Međutim, gustinu stabilne raspodele, u opštem slučaju, nije moguće odrediti analitički. Kako za dva najznačajnija slučaja, $\alpha = 1$ i $\alpha = 2$, to ipak jeste moguće, oni će biti opisani. Za $\alpha = 2$ važi da je $p(\omega_j \mid \beta) = 1$, dok za $\alpha = 1$ važi sledeća posledica teoreme 4.1.0.2.

Posledica 4.2.0.2 *Uslovna raspodela skrivenog parametra ω_j za $\alpha = 1$ je*

$$p\left(\omega_j^{-1} \mid \beta_j, \lambda\right) \sim \mathcal{IG}\left(1/\lambda\left|\beta_j\right|, 1\right).$$

Dokaz: Iz izraza 4.8 sledi

$$\begin{aligned} p\left(\omega_j \mid \beta_j, \lambda\right) &\propto \frac{1}{\sqrt{2\pi\omega_j}} \exp\left\{-\frac{1}{2}\left(\frac{\lambda^2\beta_j^2}{\omega_j}+\omega_i\right)\right\} \\ &\sim \mathcal{GIG}\left(\frac{1}{2}, 1, \lambda^2\beta_j^2\right). \end{aligned}$$

Primenom iste osobine kao u prethodnoj posledici dobija se tražena uslovna raspodela. □

4.3 Konstrukcija algoritama za ocenjivanje parametara

Uslovne raspodele izvedene u prethodnom odeljku mogu biti upotrebljene za konstruisanje EM ili MCMC algoritma za ocenjivanje nepoznatog parametra β.

Ocenjivanje nepoznatih parametara EM algoritmom

Kao što je opisano u odeljku 3.3, na strani 13, EM algoritam čine dva koraka koja se smenjuju. Pretpostavimo da je parametar regularuzacije λ fiksiran.

- E-korak: $Q\left(\beta \mid \beta^{t-1}\right) = \int \log p(\beta \mid y, \nu, \omega) p\left(\nu, \omega \mid \beta^{t-1}, y\right) d\nu d\omega$
- M-korak: $\beta^t = \arg\max_\beta Q\left(\beta \mid \beta^{t-1}\right)$

Funkcija Q predstavlja očekivanje logaritma verodostojnosti i ocenjuje se u odnosu na vrednosti parametara određenim u prethodnoj iteraciji. Logaritam aposteriorne raspodele, pod uslovom poznatih skrivenih parametara, može se izvesti iz izraza 4.9:

$$\begin{aligned} \log p(\beta \mid \nu, \omega, y) = c_0(\nu, \omega, y, \lambda) - \frac{1}{2} \sum_{i=1}^{n} \frac{\left(1 + \nu_i - y_i \mathbf{x}_i^T \beta\right)^2}{\nu_i} \\ - \frac{\lambda^2}{2} \sum_{j=1}^{k} \frac{\beta_j^2}{\omega_j}, \end{aligned} \tag{4.10}$$

za pogodno odabranu konstantu c_0.

U prethodnom izrazu učestvuju ν, ν^{-1} i ω^{-1}. Kako je Q funkcija od β, parametar λ može biti uvučen pod konstantu c_0, jer ne stoji uz β . Dakle bitan deo jednačine 4.10 je linearna funkcija od ν^{-1} i ω^{-1}, pa se funkcija $Q\left(\beta \mid \beta^{t-1}\right)$ može dobiti ako se u jednačini 4.10 ν^{-1} i ω^{-1} zamene njihovim uslovnim očekivanjima.

$$\begin{aligned} Q\left(\beta \mid \beta^{t-1}\right) &= \int c_1(\nu, \omega, y, \lambda) p\left(\nu, \omega \mid \beta^{t-1}, y\right) d\nu d\omega \\ &- \frac{1}{2} \int \sum_{i=1}^{n} \left(-2 y_i \mathbf{x}_i^T \beta \nu^{-1} - 2 y_i \mathbf{x}_i^T \beta + (y_i \mathbf{x}_i^T \beta)^2 \nu^{-1}\right) p\left(\nu, \omega \mid \beta^{t-1}, y\right) d\nu d\omega \\ &- \int \frac{\lambda^2}{2} \sum_{j=1}^{k} \frac{\beta_j^2}{\omega_j} p\left(\nu, \omega \mid \beta^{t-1}, y\right) d\nu d\omega. \\ &= c_2(y, \lambda, \beta^{t-1}) - \frac{1}{2} \sum_{i=1}^{n} \left(-2 y_i \mathbf{x}_i^T \beta \hat{\nu}_i^{-1(t)} - 2 y_i \mathbf{x}_i^T \beta + (y_i \mathbf{x}_i^T \beta)^2 \hat{\nu}_i^{-1(t)}\right) \\ &- \frac{\lambda^2}{2} \sum_{j=1}^{k} \beta_j^2 \hat{\omega}_j^{-1(t)} \\ &= \log p(\beta \mid y, \hat{\nu}^{-1(t)}, \hat{\omega}^{-1(t)}), \end{aligned}$$

gde je $\hat{\nu}_i^{-1(t)} = E\left(\nu_i^{-1} \mid y_i, \beta^t\right)$, a $\hat{\omega}_j^{-1(t)} = E\left(\omega_j^{-1} \mid y, \beta^t, \alpha\right)$. Na osnovu posledice 4.2.0.1 i osobina inverzne Gausove raspodele važi

$$\hat{\nu}_i^{-1(t)} = \left|1 - y_i \mathbf{x}_i^T \beta^t\right|^{-1}.$$

Očekivanje od ω^{-1} zavisi i od α. Za $\alpha = 2$ važi $p(\omega_j \mid \beta) = 1$, pa je $\omega_j = 1$. Za $\alpha \in (0, 2)$ rezultat se izvodi iz sledeće posledice teoreme 4.1.0.2.

Posledica 4.3.0.1 *Neka je* $0 < \alpha < 2$. *Ako je* $\beta_j^t = 0$ *tada je* $\hat{\omega}_j^{-1(t)} = E\left(\omega_j^{-1} \mid y, \beta^t, \alpha\right) = \infty$. *U suprotnom*

$$\hat{\omega}_j^{-1(t)} = \alpha \left|\beta_j^t\right|^{\alpha-2} \lambda^{\alpha-2}.$$

Dokaz: Iz teoreme 4.1.0.2 sledi:

$$p\left(\beta_j \mid \alpha\right) = \int_0^\infty \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right) p\left(\omega_j \mid \alpha\right) d\omega_j, \tag{4.11}$$

gde je, na osnovu 4.6, $p\left(\beta_j \mid \alpha\right) \propto \exp\left(-\left|\lambda\beta_j\right|^\alpha\right)$. Dalje važi

$$\frac{\partial \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right)}{\partial \beta_j} = \frac{-\lambda^2 \beta_j}{\omega_j} \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right).$$

Dakle, za $\beta_j \neq 0$ jednačina 4.11 može se diferencirati po β_j:

$$\alpha\lambda^\alpha \left|\beta_j\right|^{\alpha-1} p\left(\beta_j \mid \alpha\right) = \int_0^\infty \phi\left(\beta_j \mid 0, \lambda^{-2}\omega_j\right) p\left(\omega_j \mid \alpha\right) \frac{\lambda^2 \beta_j}{\omega_j} d\omega_j.$$

Deljenjem leve i desne strane sa $p(\beta_j \mid \alpha)$ dobija se:

$$\alpha\lambda^\alpha \left|\beta_j\right|^{\alpha-1} = \lambda^2 \beta_j \int_0^\infty \frac{1}{\omega_j} \frac{p\left(\beta_j, \omega_j \mid \alpha\right)}{p\left(\beta_j \mid \alpha\right)} d\omega_j = \lambda^2 \beta_j E\left(\omega_j^{-1} \mid \beta_j, \alpha\right),$$

odakle sledi

$$\hat{\omega}_j^{-1(t)} = E\left(\omega_j^{-1} \mid \beta_j, \alpha\right) = \alpha \left|\beta_j^t\right|^{\alpha-2} \lambda^{\alpha-2}.$$

□

Ovime je funkcija $Q\left(\beta \mid \beta^{t-1}\right)$ potpuno određena, tj. E - korak je završen, nakon čega sledi M - korak, tj. pronalaženja vrednosti parametra β koje maksimizuje Q. Kako je

$$Q\left(\beta \mid \beta^{t-1}\right) = \log p(\beta \mid y, \hat{\nu}_i^{-1(t)}, \hat{\omega}_i^{-1(t)}),$$

važi

$$\arg\max_\beta \log p(\beta \mid y, \hat{\nu}_i^{-1(t)}, \hat{\omega}_i^{-1(t)}) = \arg\max_\beta p(\beta \mid y, \hat{\nu}_i^{-1(t)}, \hat{\omega}_i^{-1(t)}),$$

a kako je $p(\beta \mid y, \hat{\nu}_i^{-1(t)}, \hat{\omega}_i^{-1(t)}) \sim \mathcal{N}(b, B)$, maksimum se dostiže u vrednosti očekivanja, pa je

$$\beta^t = \hat{b}.$$

Algoritam:

- Odrediti inicijalnu vrednost nepoznatog parametra β^0.
- E - korak: Korišćenjem parametra iz prethodne iteracije β^{t-1}, iztačunati:

$$\hat{\nu}_i^{-1(t)} = \left|1 - y_i \mathbf{x}_i^T \beta^{t-1}\right|^{-1}, \qquad i = 1, ..., n$$

$$\hat{\omega}_j^{-1(t)} = \alpha \left|\beta_j^{t-1}\right|^{\alpha-2} \lambda^{\alpha-2}, \qquad j = 1, ..., k$$

$$\hat{N}^{-1(t)} = diag(\hat{\nu}_i^{-1(t)})$$

$$\hat{\Omega}^{-1(t)} = diag(\hat{\omega}_i^{-1(t)})$$

- M - korak: Izračunati β^t:

$$\beta^t = \hat{b} = \left(\lambda^2 \hat{\Omega}^{-1(t)} + \mathbf{X}^T \hat{N}^{-1(t)} \mathbf{X}\right)^{-1} \mathbf{X}^T \left(\mathbf{1} + \hat{\nu}^{-1(t)}\right).$$

- E i M korake ponavljati do konvergencije ili do unapred zadatog broja iteracija.

Jedno značajno svojstvo EM algoritma je monotonost pseudo aposteriorne raspodele parametra β, pod uslovom opaženih podataka. Tačnije, za niz ocena parametra $\beta^1, \beta^2, \beta^3, ...$ važi $p\left(\beta^{t-1} \mid \lambda, \alpha, y\right) \leq p\left(\beta^t \mid \lambda, \alpha, y\right)$. Ovo svojstvo može biti korisno za posmatranje konvergencije niza ocena prilikom treniranja modela.

U slučaju da neka od vrednosti ν^{-1} ili ω^{-1} dostigne izuzetno visoke vrednosti, EM algoritam postaje numerički nestabilan. Štaviše, takvo ponašanje je očekivano za algoritam koji funkcioniše ispravno. Kada je $\omega^{-1} = \infty$, tada je $\beta_j = 0$. U tom slučaju odgovarajuća kolona može jednostavno biti izostavljena iz podataka. Nasuprot tome, kada je $\nu^{-1} = \infty$, tada je važi da je $y_i \beta^T \mathbf{x}_i = 1$, tj. opservacija i je baš potporni vektor. Ovi slučajevi komplikuju ocenjivanje parametara EM algoritmom, što može biti potencijalna mana ovog pristupa.

U nekim modelima postoji više nepoznatih parametara koje je potrebno odrediti. Na primer, parametar regularizecije λ je moguće oceniti iz podataka zajedno da ostalim parametrima modela. Za ocenjivanje parametara na ovakav način može se koristiti generalizacija EM algoritma, algoritam uslovne maksimizacije očekivanja, ECM (eng. *expectation-conditional maximization algorithm*) (v. [12]). Ovaj algoritam menja M korak EM algoritma nizom koraka, kod kojih se svakim korakom funkcija Q maksimizuje po jednom od parametara, pod uslovom da su svi ostali

parametri poznati (CM korak). U radu [11] je pokazano da ovaj algoritam konvergira brže ukoliko se maksimizacija funkcije Q zameni maksimizaciom aposteriorne raspodele (CME korak). Ovaj algoritam je nazvan $ECME$ i za njega, kao i za ECM algoritam važi isto svojstvo monotonosti kao i kod EM algoritma.

Za ocenjivanje parametra regularizacije λ zajedno sa β potrebno je odrediti aposteriornu raspodelu $p(\lambda \mid \beta, \alpha)$ koja će se koristiti u CME koraku. Za apriornu raspodelu parametra λ, zgodno je uzeti $\gamma(a, b)$ raspodelu, jer je ona konjugovana sa generalizovanom normalnom raspodelom iz jednačine 4.6

$$p\left(\lambda^{\alpha}\right) \propto \left(\lambda^{\alpha}\right)^{a-1} \exp\left(-b\lambda^{\alpha}\right).$$

Na osnovu 4.6 važi

$$p(\beta \mid \lambda, \alpha) \propto \lambda^{k} \exp\left(-\sum_{i=1}^{k} |\lambda\beta_j|^{\alpha}\right).$$

Odakle, za fiksirano α, sledi

$$\begin{aligned} p(\lambda \mid \beta) \propto p(\beta \mid \lambda)p(\lambda) &\propto \lambda^{k} \exp\left(-\sum_{i=1}^{k} |\lambda\beta_j|^{\alpha}\right) \left(\lambda^{\alpha}\right)^{a-1} \exp\left(-b\lambda^{\alpha}\right) \\ &= \left(\lambda^{\alpha}\right)^{a+k/\alpha-1} \exp\left(-\lambda^{\alpha}\left(b + \sum_{i=1}^{k} |\beta_j|^{\alpha}\right)\right). \end{aligned}$$

Dakle,

$$p(\lambda \mid \beta) \sim \gamma\left(a + \frac{k}{\alpha}, b + \sum_{i=1}^{k} |\beta_j|^{\alpha}\right). \tag{4.12}$$

Kod ECME algoritma E korak je isti kao i kod EM algoritma,s' tim što λ više nije fiksiran već se uzima iz prethodne iteracije, dok se M korak menja CME korakom. Za ocenu parametra λ uzima se moda, kao tačka koja maksimizuje aposteriornu raspodelu.

Algoritam:

- Odrediti inicijalnu vrednost nepoznatih parametra λ^0 i β^0.
- E - korak: Korišćenjem parametra iz prethodne iteracije λ^{t-1} i β^{t-1}, iztačunati:

$$\hat{\nu}_i^{-1(t)} = \left|1 - y_i \mathbf{x}_i^T \beta^{t-1}\right|^{-1}, \qquad i = 1, ..., n$$

$$\hat{\omega}_j^{-1(t)} = \alpha \left|\beta_j^{t-1}\right|^{\alpha-2} (\lambda^{t-1})^{\alpha-2}, \qquad j = 1, ..., k$$
$$\hat{N}^{-1(t)} = diag(\hat{\nu}_i^{-1(t)})$$
$$\hat{\Omega}^{-1(t)} = diag(\hat{\omega}_i^{-1(t)})$$

- CME - korak: Izračunati β^t i λ^t:

$$\beta^t = \left(\lambda^{2(t-1)}\hat{\Omega}^{-1(t)} + \mathbf{X}^T\hat{N}^{-1(t)}\mathbf{X}\right)^{-1} \mathbf{X}^T\left(\mathbf{1} + \hat{\nu}^{-1(t)}\right)$$

$$\lambda^t = \frac{a + \frac{k}{\alpha} - 1}{b + \sum_{i=1}^{k}\left|\beta_j^t\right|^{\alpha}}$$

- E i CME korake ponavljati do konvergencije ili do unapred zadatog broja iteracija.

Ocenjivanje nepoznatih parametara MCMC algoritmom

Reprezentacija mešavinom normalnih raspodela, opisana u odeljku 4.1, omogućava korišćenje Gibsovog algoritma za ocenjivanje vrednosti nepoznatih parametara. Za svaki od nepoznatih parametara, potrebno je odrediti raspodele uslovljene vrednostima svih ostalih parametara koji učestvuju u modelu. One su izvedene u odeljku 4.2 i imamo:

$$p\left(\nu_i^{-1} \mid \beta, y_i\right) \sim \mathcal{IG}\left(\left|1 - y_i\mathbf{x}_i\beta\right|^{-1}, 1\right),$$
$$p\left(\omega_j^{-1} \mid \beta_j, \lambda\right) \sim \mathcal{IG}\left(1/\lambda\left|\beta_j\right|, 1\right), \qquad \alpha = 1,$$
$$\omega = 1, \qquad \alpha = 2,$$
$$p(\beta \mid y, \nu^{-1}, \omega^{-1}) \sim \mathcal{N}(b, B).$$

Algoritam:

- Odrediti inicijalnu vrednost parametara $\nu^{-1(0)}$, $\omega^{-1(0)}$ i β^0
- Za $t = 0, ..., N-1$:

$$\nu^{-1(t+1)} \sim p(\nu^{-1} \mid \beta^{(t)}, y), \qquad i = 1, ..., n$$
$$\omega^{-1(t+1)} \sim p(\omega^{-1} \mid \beta^{(t)}, y), \qquad i = 1, ..., k$$
$$\beta^{(t+1)} \sim p(\nu^{-1} \mid \lambda^{(t)}, \nu^{(t)}, y)$$

Ovim algoritmom je, dodavanjem jednog koraka, moguće izvući uzorak i za parametar regularizacije λ iz odgovarajuće uslovne raspodele. Na isti način kao i kod 4.12 imamo

$$p(\lambda \mid \beta) \sim \gamma \left(a + \frac{k}{\alpha}, b + \sum_{i=1}^{k} |\beta_j|^{\alpha} \right).$$

Kako su eksplicine forme uslovnih raspodela, koje učestvuju u algoritmu, poznate, za ocenu parametra možemo uzeti uzoračku sredinu, kao u 3.2.

Glava 5

Primene

Oba pristupa u ocenjivanju parametara modela, EM i MCMC algoritmom, testirana su nad različitim skupovima podataka i rezultati su upoređivani nad svakim od njih. Kodovi su pisani u programskom jeziku R. Odabrani skupovi podataka su oni koji se često koriste u literaturi za poređenje kvaliteta modela klasifikacije.

1. Podaci *Spam* sadrže 4601 mejl koji je klasifikovan na one koji jesu i koji nisu nepoželjna pošta. Sadrže 57 promenljivih od kojih 54 predstavljaju frekvencije pojedinih reči, brojeva ili specijalnih znakova u tekstu. Ostale 3 promenljive su prosečan broj, najduži niz i ukupan broj velikih slova u tekstu mejla (v. [10]).

2. Podaci *BreastCancer* opisuju tumore koji se klasifikuju na maligne i benigne. Podaci sadrže 699 opservacija i 10 promenljivih koje opisuju različite karakteristike tumora (v. [16]).

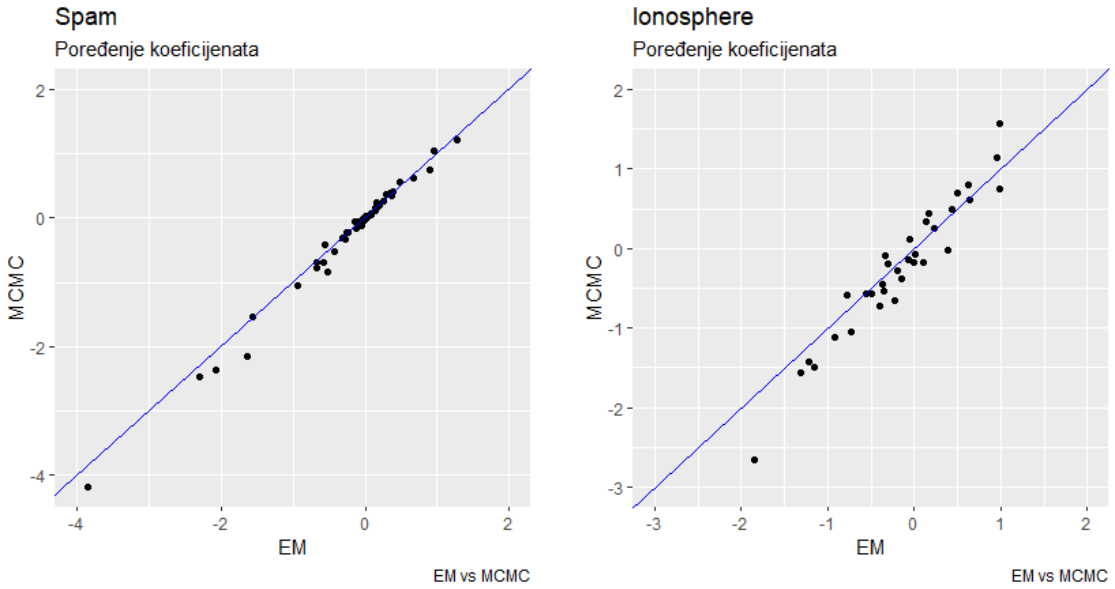
3. Podaci *Ionosphere* prikupljeni su od strane sistema u mestu *Goose Bay, Labrador*. Sistem se sastoji od antena koje emituju elektromagnetne talase koji gađaju elektrone u jonosferi. Talasi koji su se odbili detektuju prisustvo određene strukture u jonosferi, a opservacije koje odgovaraju tim slučajevima klasifikuju se pozitivno. Podaci sadrže 351 opservaciju i opisani su pomoću 34 varijable (v. [21]).

4. Podaci *PimaIndiansDiabetes* sadrže dijagnostičke podatke 768 žena Pima indijskog porekla. Klasifikuju se po prisustvu dijabetesa i opisuju se pomoću 9 promenljivih (v. [15]).

5. Podaci *Sonar* sadrži 208 opservacija i 60 promenljivih koje predstavljaju zvučne signale koji se klasifikuju po tome da li su se dobili od metal ili od stenu. (v. [22]).

5.1 Rezultati

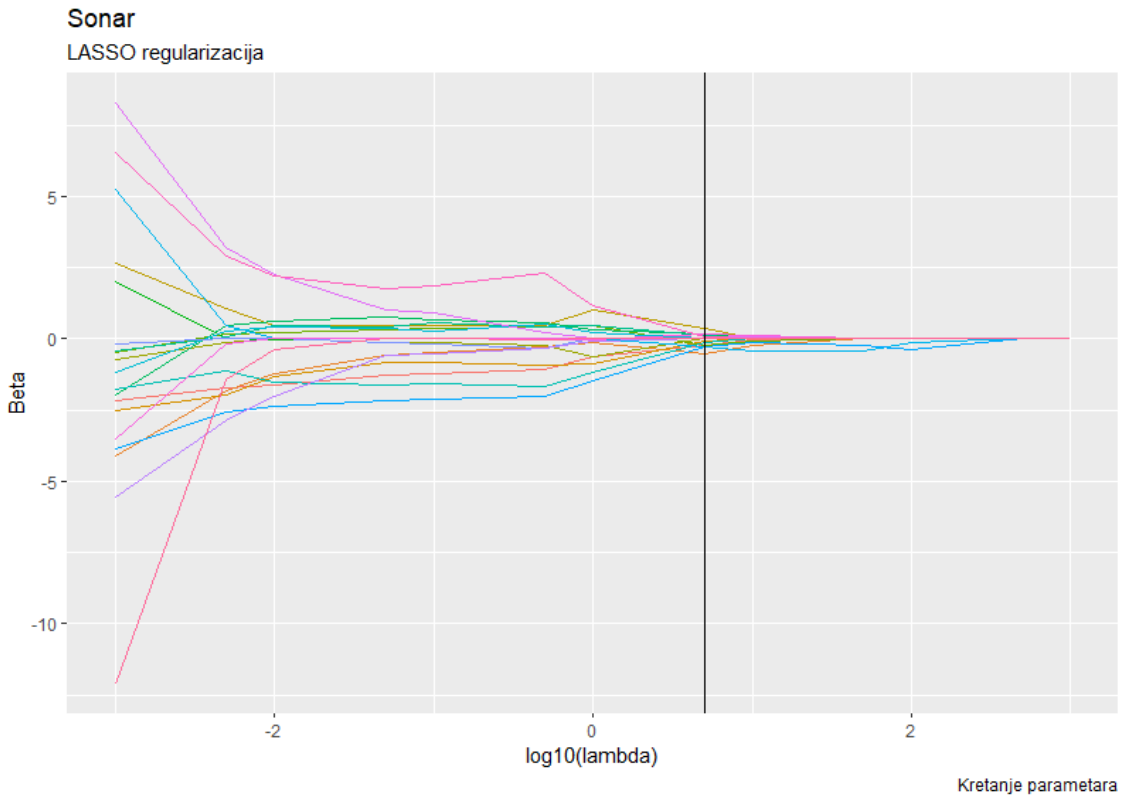
Iako su EM i MCMC algoritmi dva različita pristupa u ocenjivanju parametara, model koji ti parametri određuju je isti, pa bi i koeficijenti modela, tj. ocene traženih parametara, dobijeni pomoću oba pristupa trebalo da budu približno jednaki. Na sledećem grafiku je dato poređenje koeficijenata modela dobijenih EM i MCMC algoritmom, za iste vrednosti hiperparametara $\alpha = 2$, $\lambda = 1$.



Slika 5.1: Parametri modela ocenjeni EM i MCMC algoritmom

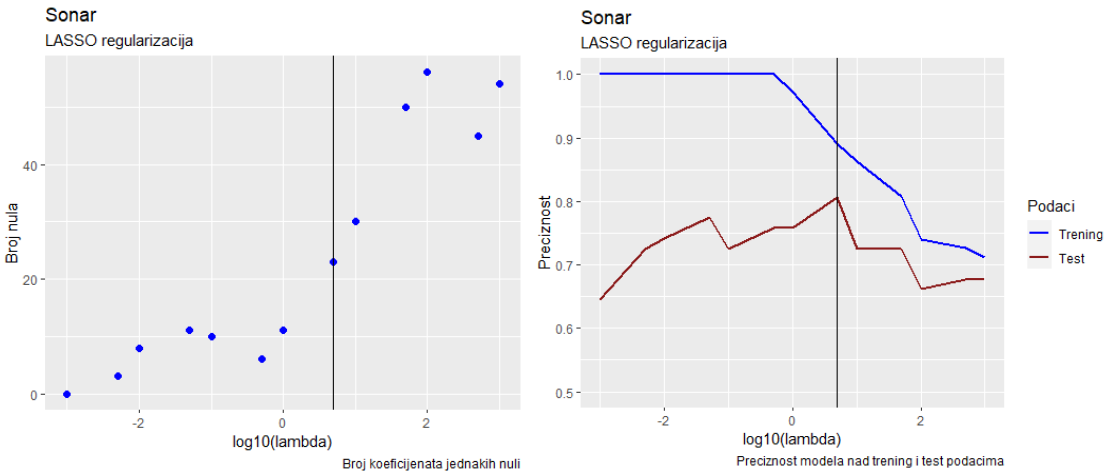
Vrednost hiperparametra $\alpha = 1$ odgovara *LASSO* regularizaciji (eng. *Least Absolute Shrinkage and Selection Operator*). Kao što je već pomenuto u poglavlju 2, a na osnovu optimizacionog izraza 2.1, vrednost hiperparametra λ kontroliše nivo regularizacije modela. Na sledećim graficima prikazano je kretanje parametara modela u zavisnosti od nivoa regularizacije.

Takođe, za razliku od grebene regresije, *LASSO* ima osobinu da koeficijente modela može dovesti tačno u nulu, čime se ujedno dobija i proređen model, tj. parametri modela, koji su manje značajni, se izostavljaju. Na podacima *Sonar* prikazano je



Slika 5.2: Zavisnost parametara modela od nivoa LASSO regularizacije

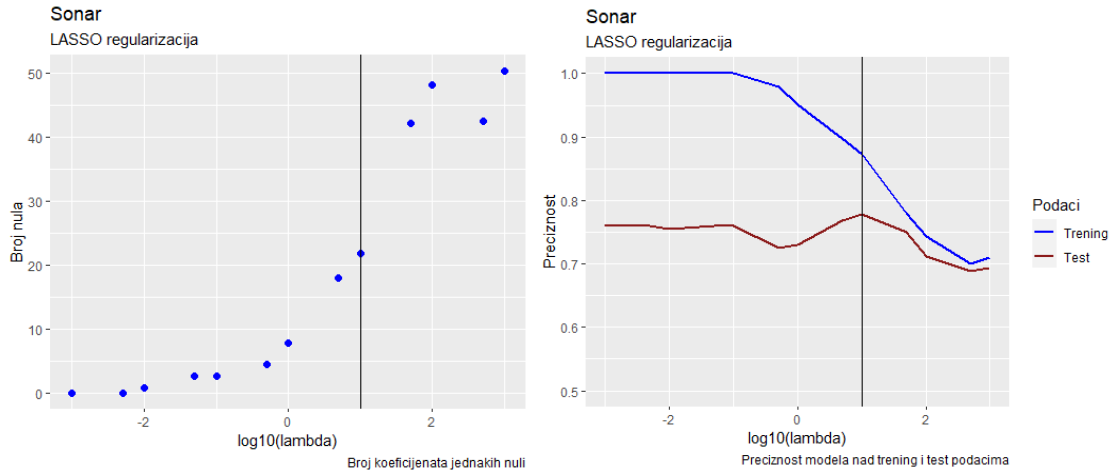
kretanje broja parametara koji su isključeni iz modela, kao i tačnost predviđanja modela na skupovima za obučavanje i testiranje.



Slika 5.3: Tačnost i proređenost modela

Vertikalna linija na graficima označava ono λ za koje su performanse modela najbolje. U ovom slučaju to je $\lambda = 5$.

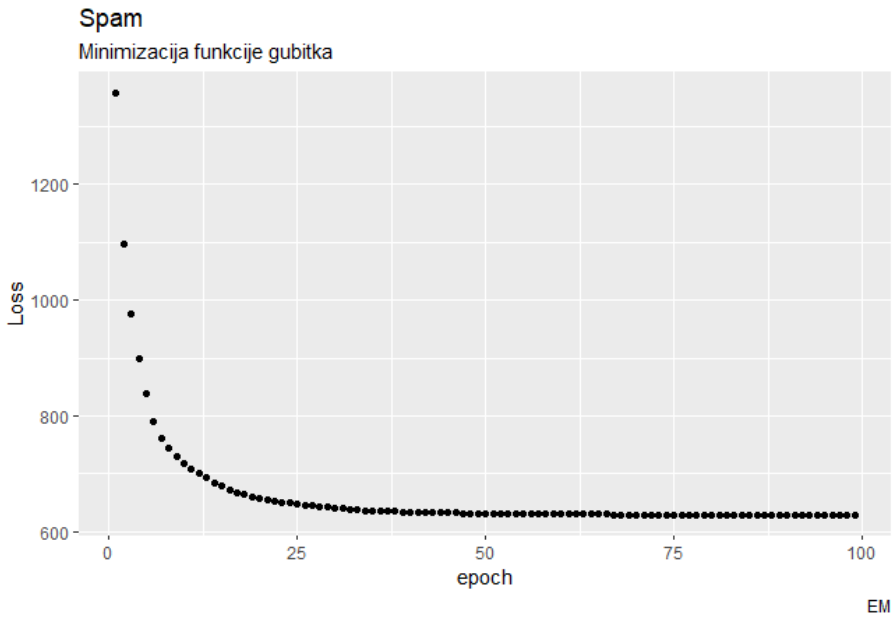
Rezultati prikazani na prethodnim graficima zavise od podele na trening i test podatke, ali su prikazani kao ilustracija rada regularizacije. Na narednim graficima prikazano je prosečno ponašanje modela nad 10 različitih podela na trening i test podatke. Sa grafika se vidi da su performanse modela u proseku ipak bolje za $\lambda = 10$.



Slika 5.4: Prosečna tačnost i proređenost modela

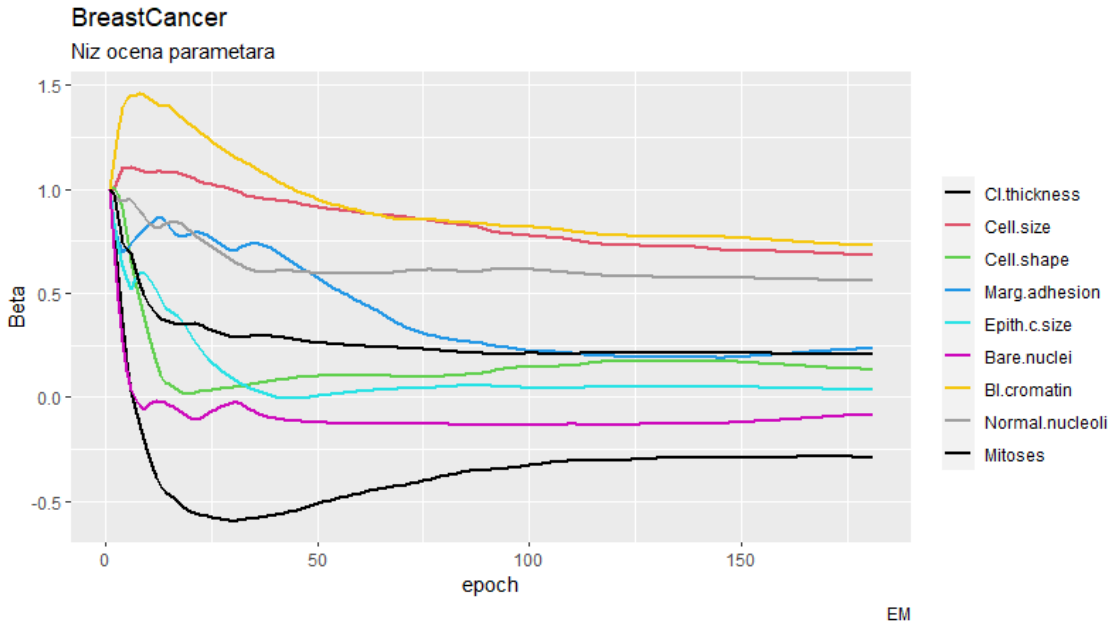
Primenom EM algoritma dobija se niz ocena parametara $\beta^1, \beta^2, \beta^3, \ldots$ za koji važi $p\left(\beta^{t-1} \mid \lambda, \alpha, y\right) \leq p\left(\beta^t \mid \lambda, \alpha, y\right)$. Kako važi i da je $p(\beta \mid \lambda, \alpha, y) \propto \exp\left(-d_\alpha(\beta, \lambda)\right)$ (4.2), gde je $d_\alpha(\beta, \lambda)$ optimizacioni izraz 4.1, imamo da niz ocena optimizacionog izraza $\{d_\alpha(\beta^{(t)}, \lambda)\}_{t=1}^{\infty}$ monotono opada. Ova osobina može biti korisna za posmatranje kovergencije niza ocena, a može se koristiti i kao jedna od provera ispravnosti implementacije algoritma.

Sledeći grafik pokazuje jedan mogući tok treniranja modela EM algoritmom za parametre $\alpha = 2$, $\lambda = 1$ nad *Spam* podacima.



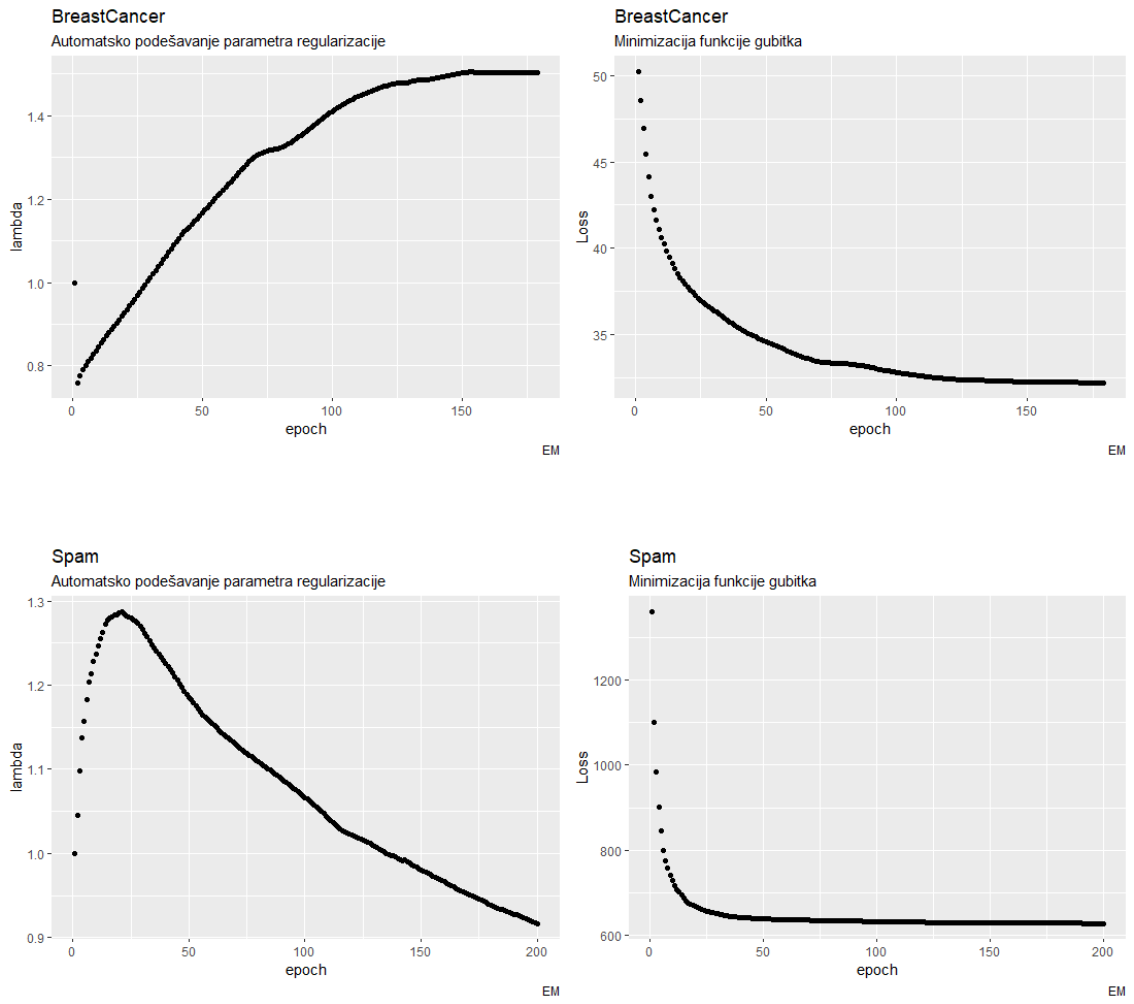
Slika 5.5: Monotonost funkcije gubitka

Tokom treniranja modela, svakom iteracijom ažuriraju se ocenjene vrednosti parametara modela. Na narednom grafiku je primer niza ocena parametara modela, za inicijalne vrednosti $\beta^0 = \mathbf{1}$.



Slika 5.6: Niz ocena parametra beta

Pored parametra β, kao što je napomenuto, Bajesovim pristupom metodu potpornih vektora, moguće je i parametar regularizacije λ oceniti iz podataka, zajedno sa parametrom β. Na narednom grafiku prikazan je niz parametara regularizacije koji su ocenjivani tokom treniranja nad podacima *BreastCancer* i *Spam*, kao i funkcije gubitka u oba slučaja.



Slika 5.7: Odabir parametra lambda

Iako na prethodnim graficima, zbog različitih skala na y osi, ne deluje tako, model je nad podacima *BreastCancer* iskonvergirao posle manje od 200 iteracija, dok nad podacima *Spam* nije. U prvom slučaju regularizacija dostiže stabilan nivo $\lambda = 1.5$, dok u drugom λ ima opadajući trend i zahteva veći broj iteracija da bi se postigla konvergencija, što odgovara i većoj dimenzionalnosti podataka *Spam* u odnosu na *BreastCancer*. Ovo je ujedno i jedna od mana iterativnog pristupa treniranju modela.

Nad *Spam* podacima, koji imaju 4601 opservaciju, 500 iteracija nad 70% podataka traje skoro 5 minuta. Kako par hiljada opservacija, za današnje standarde, nije tako puno, upotreba ovog pristupa nad realim podacima postaje neefikasna. Nedavno je u radu [24] ponuđeno jedno rešenje ovog problema, što može biti tema za dalji rad.

U tabeli 5.1 data su vremena obučavanja modela u sekundama, nad 70% podataka i za maksimalno 500 iteracija.

	EM		**MCMC**			
Podaci	$\alpha = 2\ \lambda = 1$	$\alpha = 1\ \lambda = a$	$\alpha = 2\ \lambda = 1$	$\alpha = 1\ \lambda = a$	n	k
Spam	269.35	216.77	272.38	275.66	4601	57
BreastCancer	0.43	0.33	1.72	1.69	699	10
Ionosphere	0.21	1.01	1.56	1.59	351	34
PimaIndians	1.37	0.11	2.00	1.89	768	9
Sonar	0.55	0.45	1.64	1.70	208	60

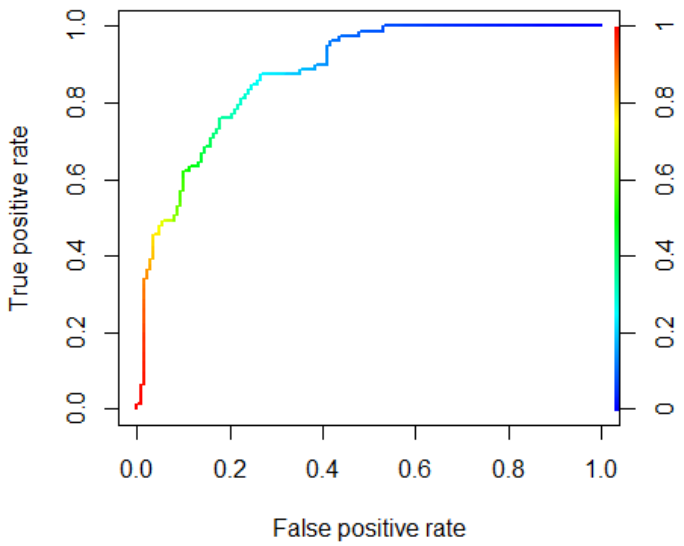
Tabela 5.1: Vreme treniranja u sekundama

Metod potpornih vektora je metod zasnovan na širokom pojasu. Mesto širokog pojasa, kao i optimalne hiperravni koja razdvaja klase, određeno je potpornim vektorima. Novim podacima koji se klasifikuju klasa je dodeljena u zavisnosti od toga sa koje strane hiperravni se nalaze, ali osnovna verzija ovog metoda ne daje nikakvu informaciju o verovatnoći pripadanja rezultujućim klasama.

Jedna od prednosti Bajesovog pristupa metodu potpornih vektora je upravo to. Imajući aposterirornu raspodelu parametara modela, na osnovu 3.1, moguće je odrediti verovatnoću pripadanja klasi.

Tako dobijene verovatnoće mogu biti vrlo korisne prilikom podešavanja modela. Na primer, moguće je promeniti prag klasifikacije sa 0.5 koji je podratumevan na neki drugi koji daje bolje rezultate. Opseg mera kvaliteta kojima model može biti evaluiran se takođe širi i na one kojima je neophodna verovatnoća. Jedan primer takve mere je AUC, koja se veoma često koristi, pogotovo za merenje kvaliteta modela treniranim nad nebalansiranim skupom podataka.

Nad podacima *PimaIndiansDiabetes* dobijen je $AUC = 0.878$ korišćenjem MCMC algoritma za ocenu parametara i uz $\alpha = 2$, $\lambda = 1$.



Slika 5.8: ROC kriva

Evaluacija modela rađena je nad skupom za testiranje, koji čini 30% podataka. Podela nije postojala prilikom preuzimanja podataka, već je nastala nasumičnim odabirom, stratifikovanim po ciljnoj promenljivoj. Odabrane mere kvaliteta su tačnost i F1 mera. Tačnost se koristi jer je najjednostavnija mera, dok F1 mera bolje odgovara neizbalansiranim podacima. Odabrane su takođe jer se mogu izračunati bez znanja o verovatnoćama pripadnosti klasama, što je važno zbog poređenja sa drugim modelima, o kojima će biti reči kasnije.

		Predviđena	
		-1	1
Prava	-1	134	17
	1	29	50

Na osnovu date matrice konfuzije, za isti model, imamo:

- Tačnost = 80%
- F1 mera = 0.685

Korišćenjem ovih mera mogu se uporediti modeli čiji parametri se ocenjuju EM i MCMC algoritmom, kao i njihove performanse za različite vrednosti hiperparametara. U narednoj tabeli prikazani su rezultati dobijeni primenom EM algoritma, sa najviše 500 iteracija, za vrednosti parametara $\alpha \in \{1, 2\}$, $\lambda \in \{1, 10, auto\}$, gde opcija *auto* označava automatsko biranje nivoa regularizacije tokom obučavanja modela.

Modeli su evaluirani unakrsnom validacijom sa 5 slojeva, a podele su rađene stratifikacijom po ciljnoj promenljivoj.

$\alpha = 1$	**Tačnost**			**F1 mera**		
Podaci	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$
Spam	0.929	0.931	0.930	0.908	0.911	0.909
BreastCancer	0.966	0.967	0.966	0.950	0.953	0.950
Ionosphere	0.880	0.878	0.860	0.810	0.804	0.772
PimaIndiansDiabetes	0.773	0.775	0.772	0.636	0.636	0.633
Sonar	0.740	0.754	0.755	0.718	0.732	0.731

$\alpha = 2$	**Tačnost**			**F1 mera**		
Podaci	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$
Spam	0.929	0.922	0.929	0.909	0.899	0.908
BreastCancer	0.966	0.969	0.966	0.950	0.955	0.950
Ionosphere	0.852	0.880	0.872	0.746	0.809	0.795
PimaIndiansDiabetes	0.773	0.769	0.773	0.636	0.632	0.636
Sonar	0.750	0.793	0.741	0.726	0.782	0.709

Tabela 5.2: Rezultati modela primenom EM algoritma

Za iste vrednosti parametara α i λ prikazani su rezultati primenom MCMC algoritma sa 500 iteracija. Nakon toga je dat uporedni prikaz oba algoritma i za svaki od njih su izdvojeni parametri modela koji daju najbolje rezultate.

$\alpha = 1$	**Tačnost**			**F1 mera**		
Podaci	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$
Spam	0.930	0.931	0.931	0.910	0.911	0.911
BreastCancer	0.966	0.966	0.966	0.950	0.951	0.950
Ionosphere	0.877	0.877	0.883	0.804	0.802	0.812
PimaIndiansDiabetes	0.776	0.771	0.773	0.638	0.630	0.634
Sonar	0.760	0.798	0.765	0.734	0.784	0.737

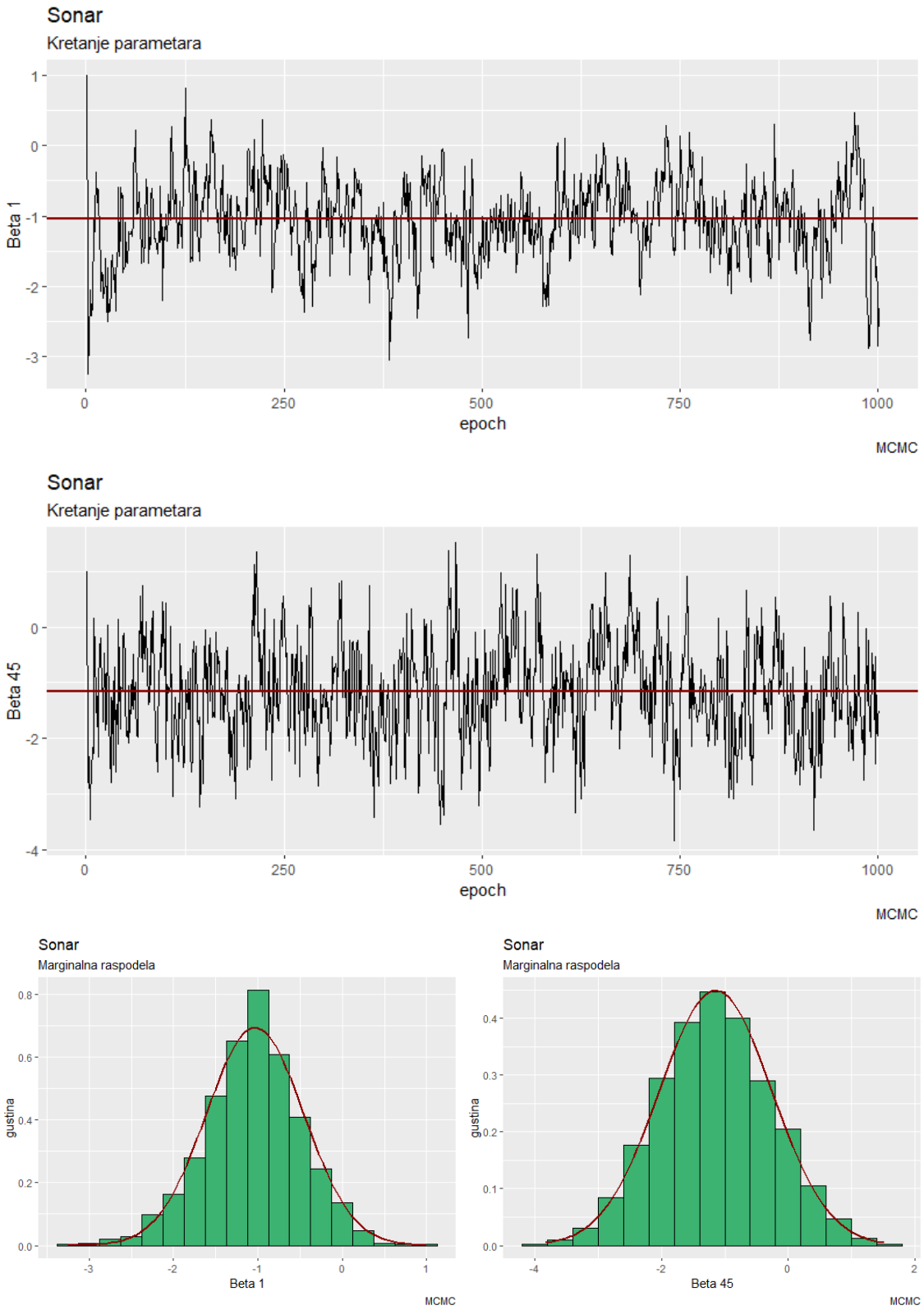
$\alpha = 2$	**Tačnost**			**F1 mera**		
Podaci	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$	$\lambda = 1$	$\lambda = 10$	$\lambda = auto$
Spam	0.929	0.922	0.930	0.909	0.898	0.910
BreastCancer	0.966	0.967	0.967	0.950	0.953	0.952
Ionosphere	0.877	0.886	0.869	0.808	0.820	0.789
PimaIndiansDiabetes	0.773	0.771	0.775	0.633	0.634	0.635
Sonar	0.774	0.803	0.784	0.753	0.793	0.757

Tabela 5.3: Rezultati modela primenom MCMC algoritma

	EM			**MCMC**		
Podaci	**Tačnost**	**F1**	α, λ	**Tačnost**	**F1**	α, λ
Spam	0.931	0.911	1, 10	0.931	0.911	1, a
BreastCancer	0.969	0.955	2, 10	0.967	0.953	2, 10
Ionosphere	0.880	0.810	1, 1	0.886	0.820	2, 10
PimaIndiansDiabetes	0.775	0.636	1, 10	0.776	0.638	1, 1
Sonar	0.793	0.782	2, 10	0.803	0.793	2, 10

Iz prethodnih tabela može se zaključiti da se model primenom oba metoda za ocenjivanje parametara, u većini slučajeva, ponaša prilično slično. Takođe, biranje parametra λ tokom obučavanja modela ne daje ništa bolje rezultate od nasumično odabranih parametara $\lambda = 1$, ili $\lambda = 10$. Jedno moguće objašnjenje bi bilo da se prilikom treniranja modela parametar λ bira tako da funkcija gubitka bude minimalna, što nije uvek garancija da će i na skupu za testiranje dati bolje rezultate jer je moguće da preteranom minimizacijom dođe do preprilagođavanja. Pouzdaniji način izbora parametra λ bi ipak bio, na primer, korišćenjem validacionog skupa, ali i ovako dobijeno λ može poslužiti kao dobar orijentir, u kom intervalu bi tražiti optimalnu vrednost.

Kod MCMC algoritma, zajednička raspodela svih nepoznatih parametara se uzorkuje i samim tim model ne mora uvek ispasti isto. Takođe se gubi i monotonost funkcije gubitka tokom obučavanja modela. Uprkos tome, za dovoljan broj iteracija, tj. za dovoljno veliki uzorak, rezultati su prilično slični onima dobijenim primenom EM algoritma, kao što je prikazano u tabelama 5.2 i 5.3 i na grafiku 5.1. Marginalne raspodele svakog od parametra se mogu oceniti iz uzorka. Na narednim graficima prikazano je kretanje dva proizvoljna koeficijenta modela, kao i njihove raspodele. Parametri modela su ocenjeni kao uzoračka sredina na osnovu uzorka obima $N = 1000$, nakon što se prvih 50 uzoraka izbaci kao $burn - in$. Model je treniran nad podacima *Sonar*, uz parametre $\alpha = 2$ i $\lambda = auto$.



Slika 5.9: Kretanje parametara β_1 i β_{45}

5.2 Poređenje sa drugim modelima

Bajesov metod potpornih vektora, u daljem tekstu *BSVM* primenjen je nad skupovima podataka opisanih na početku poglavlja 5. Nad tim podacima, model je poređen sa logističkom regresijom, linearnom i nelinearnom varijantom standardnog metoda potpornih vektora, slučajnim šumama, kao i sa klasifikacijom Gausovim procesima. U sledećim tabelama dati su rezultati primene ovih modela.

Podaci	**LR**	**SVM linearni**	**SVM nelinearni**	**GP**	**RF**	**BSVM**
Spam	0.914	0.930	0.933	0.931	**0.954**	0.931
BreastCancer	0.967	0.966	**0.971**	0.963	0.969	0.969
Ionosphere	0.880	0.875	**0.934**	0.866	0.920	0.886
PimaIndiansDiabetes	0.776	0.773	0.771	0.768	0.768	**0.776**
Sonar	0.717	0.769	**0.866**	0.813	0.856	0.803

Tabela 5.4: Tačnost klasifikacije

Podaci	**LR**	**SVM linearni**	**SVM nelinearni**	**GP**	**RF**	**BSVM**
Spam	0.892	0.910	0.914	0.911	**0.941**	0.911
BreastCancer	0.952	0.950	**0.959**	0.946	0.955	0.955
Ionosphere	0.815	0.797	**0.900**	0.773	0.886	0.820
PimaIndiansDiabetes	0.642	0.636	0.631	**0.644**	0.637	0.638
Sonar	0.692	0.746	**0.849**	0.787	0.837	0.793

Tabela 5.5: F1 mera

Poslednja kolona predstavlja najbolji rezultat BSVM modela izvučen iz tabela 5.2 i 5.3. Na svim podacima, osim na *PimaIndiansDiabetes* nelinearni modeli daju nešto bolje rezultate. Najbolje se pokazao nelinearni metod potpornih vektora, mada na ovim podacima ni ostali modeli nisu drastično lošiji.

BSVM daje rezultate najsličnije linearnom metodu potpornih vektora, što je i očekivano, jer su to suštinski isti modeli, samo razvijeni različitim pristupima. Uporedivnost ovih rezultata ukazuje na to da algoritam zaista radi i da je korektno implementiran.

U tabelama 5.4 i 5.5, za linearni metod potpornih vektora, prikazani su rezultati sa podrazumevanim parametrom regularizacije $\lambda = 1$. Mogućnost Bajesovog metoda potpornih vektora, da bira parametar λ tokom obučavanja, zajedno sa ostalim

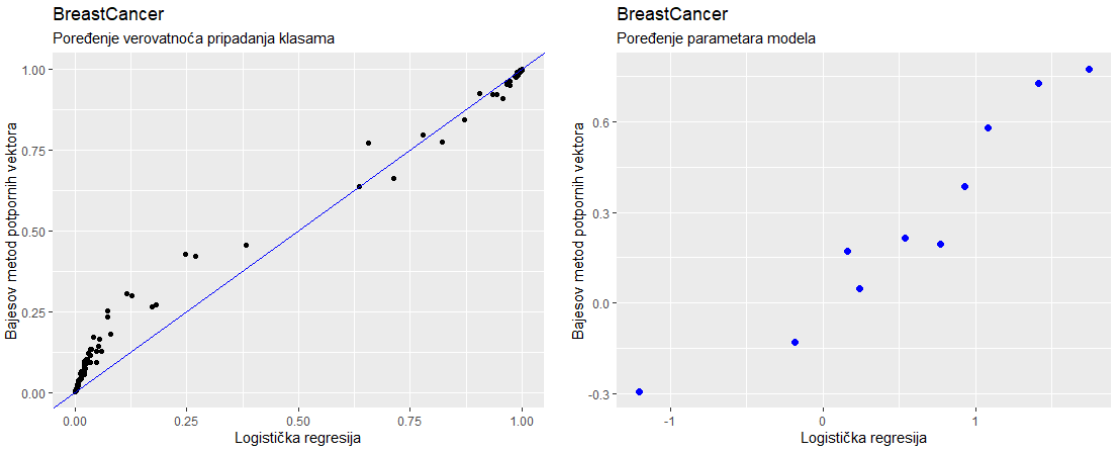
parametrima modela, mogla bi se uporediti sa vrednostima koje se kod linearnog metoda potpornih vektora pokažu kao optimalne. Izdvojeno je 70% podataka za obučavanje, stratifikovanih po ciljnoj promenljivoj, nad kojima je treniran $BSVM$, sa odabirom nivoa regularizacije. Nad istim podacima za obučavanje, unakrsnom validacijom odabran je optimalan parametar regularizacije za linearan SVM, nakon čega je za tako odabrano λ model istreniran na celom skupu za obučavanje. Oba modela su evaluirana na preostalih 30% podataka. Rezultati primene modela, kao i parametri λ odabrani na oba načina, prikazani su u tabeli 5.6.

	BSVM			**SVM**		
Podaci	**Tačnost**	**F1 mera**	λ	**Tačnost**	**F1 mera**	λ
Spam	0.922	0.900	2.012	0.920	0.898	100
BreastCancer	0.957	0.931	2.103	0.957	0.931	0.05
Ionosphere	0.857	0.800	0.002	0.895	0.836	0.5
PimaIndiansDiabetes	0.796	0.680	2.566	0.804	0.690	0.5
Sonar	0.758	0.706	2.045	0.758	0.694	0.05

Tabela 5.6: Regularizacija

Na ovim konkretnim podacima regularizacija nema previše uticaja, pa se ne može utvrditi da li odabir parametra λ Bajesovim metodom potpornih vektora ima praktičnog značaja.

Logistička regresija takođe ocenjuje verovatnoće pripadanja klasama, pa je moguće i na taj način uporediti rezultate. Naredni grafici porede verovatnoće dobijene BSVM modelom i logističkom regresijom, kao i parametre tih modela.



Slika 5.10: Logistička regresija vs BSVM

Sa levog grafika može se videti da, iako BSVM dodeljuje nešto veće verovatnoće pripadanja pozitivnoj klasi nego logistička regresija, ako se za prag uzme 0.5, dodeljene klase se svim opservacijama iz skupa za testiranje poklapaju.

U BSVM model uključena je i regularizacija, kojoj u ovom slučaju odgovaraju parametri $\alpha = 2$, $\lambda = 1$. Zbog toga su parametri ocenjeni BSVM modelom bliži nuli od onih dobijenih logističkom regresijom, ali sa desnog grafika se vidi da su međusobno uglavnom proporcionalni.

Glava 6

Zaključak

Nalaženje minimuma funkcija gubitka metoda potpornih vektora 4.1 vrši se algoritmima koji su uglavnom malo poznati statističarima. Bajesovim pristupom optimizaciji, ovaj problem se rešava metodama i algoritmima koji se u statistici često koriste. Ipak, obučavanje modela ovim pristupom može biti prilično sporo na podacima veće dimenzije, na šta ukazuje tabela 5.1.

Jedna od prednosti ovog pristupa je što pruža direktnu informaciju o verovatnoći pripadnosti klasama, što omogućava bolju evaluaciju i daje veću fleksibilnost u podešavanu modela. Primenom MCMC algoritma dobija se uvid u zajedničku raspodelu svih nepoznatih parametara.

Kao što je prikazano u tabelama 5.4 i 5.5, rezultati dobijeni primenom ovog metoda slični su rezultatima dobijenim drugim popularnim metodama binarne klasifikacije.

Glava 7

Dodatak

7.1 Kodovi

```
library(dplyr)
library(MLmetrics)
library(mlbench)
library(kernlab)
library(MASS)


trainBSVM<-function(X, y, alpha, estimator, hyper, nepoch = 1000){

  # X - training data - converted to (n,k) matrix including
    intercept
  # y - target values - takes binary, logical or +/-1 values -
    converted to +/-1
  # alpha - type of regularization - 1 (lasso) and 2 (ridge) are
    supported
  # estimator - EM or MCMC
  # hyper - regularization constant (int) or 'auto' - tuned
    automatically

  # nepoch - number of iterations for estimator - default = 1000



  column_names_orig<-c('Intercept',names(X))
  column_names<-column_names_orig

```

```
23  X<-as.matrix(X)
24  X<-scale(X)
25
26  center<-attr(X,'scaled:center')
27  scale<-attr(X,'scaled:scale')
28
29  n<-nrow(X)
30  X<-cbind(rep(1,n),X) #additional column for intercept
31  k<-ncol(X)
32
33
34
35  if(is.logical(y) | all(y %in% c(0,1))){
36    y<-y*2-1
37  }else if(! all(y %in% c(-1,1))){
38    stop("invalid y - should be binary, logical or +/-1")
39  }
40
41
42
43  if(! alpha %in% c(1,2)){
44    stop("Unsupported value of alpha. Put 1 for lasso or 2 for
     ridge.")
45  }
46
47
48
49  if(estimator=="MCMC"){
50    mcmc<-1
51  }else if(estimator=="EM"){
52    mcmc<-0
53  }else{
54    stop("Unsupported estimator. Use MCMC or EM.")
55  }
56
57
58
59  if(is.numeric(hyper)){
60    lambda<-rep(hyper,nepoch+1)
61  }else if(hyper=='auto'){
62    lambda<-rep(0,nepoch+1)
63    lambda[1]<-1 #default initial value
```

```
64  }else{
65    stop("hyper should be numeric or 'auto'")
66  }
67
68
69  ################################################################
70
71
72  #initial values:
73
74  burnin<-50            #number of initial samples which are
     discarded - MCMC
75  fvals<-rep(0,nepoch) #value of objective function for each
     iteration
76  mfval<-rep(0,nepoch) #value of objective function - MCMC
77  mb<-rep(0,k)          #optimal beta - MCMC
78  ib<-rep(1,k)          #initial beta
79  invomega<-rep(1,k)    #latent variable
80  b<-matrix(0, nrow = nepoch+1, ncol = k) #saving paths for beta
81  minf<-Inf             #min(fvals)
82  cnvg<-0               #convergence indicator
83  ind<<-0               #EM is unstable indicator
84  lasso<-c()            #keeps column names discarded by LASSO
85
86
87
88  #estimate
89
90  b[1,]<-ib
91
92  for(i in 1:nepoch){
93    if(i %% (nepoch %/% 10) ==0){
94      message(paste(i/(nepoch %/% 10),'0%',sep = ''))
95    }
96
97
98    #E step
99
100   invnu<-est_invnu(X,y,b[i,],mcmc) #Step 1:
101
102   if(alpha!=2){
103     invomega<-est_invomega(alpha,b[i,],lambda[i],mcmc,k) #Step 2:
```

```
104     }
105
106
107     #M step / Step 3:
108
109
110     #LASSO
111     izbaciti<-which(invomega>100000000000000)
112
113     if(length(izbaciti)>0 & mcmc==0){
114       lasso<-c(lasso,column_names[izbaciti])
115       if(sum(is.na(lasso))>0){
116         break
117       }
118       column_names<-column_names[-izbaciti]
119       X<-X[,-izbaciti]
120       b<-b[,-izbaciti]
121       invomega<-invomega[-izbaciti]
122       k<-k-length(izbaciti)
123     }
124
125
126     tryCatch(b[i+1,]<-est_b(X,y,invnu,invomega,lambda[i],mcmc,k),
127              error=function(e){
128                ind<<-1
129                print("EM is unstable - System is computationally
    singular.")
130              })
131     if(ind==1){
132       b[i+1,]<-b[i,]
133       cnvg<-10
134     }
135
136     #CME step / Step 4:
137
138     if(hyper=='auto'){
139       lambda[i+1]<-est_lambda(alpha,b[i+1,],mcmc,2,1,k)
140     }
141
142
143     #average over sample paths for MCMC
144     if(i>burnin & mcmc){
```

```
145       mb<-mb+(b[i+1,]-mb)/(i-burnin)
146       mfval[i]<-objective(X,y,mb,lambda[i+1],alpha)
147     }else{
148       mfval[i]<-NA
149     }
150 
151 
152     #loss function
153     fvals[i]<-objective(X,y,b[i+1,],lambda[i+1],alpha)
154 
155 
156     if(abs(minf-fvals[i])<=0.001){
157       cnvg<-cnvg+1
158     }
159     if(cnvg>=10){
160       if(ind==0){
161         print(paste("Convergence reached after",i,"iterations."))
162       }
163       break
164     }
165     minf<-min(fvals[i],minf)
166 
167   }
168 
169   if(!mcmc){
170 
171     y_pred<-(X%*%b[i+1,]>=0)*2-1
172 
173     coef0<-which(column_names_orig %in% lasso)
174 
175     mb<-c(b[i+1,],rep(0,length(coef0)))
176     id<-c(seq_along(b[i+1,]), coef0+0.5-c(1:length(coef0)))
177 
178     mb<-mb[order(id)]
179   }else{
180     y_pred<-(X%*%mb>=0)*2-1
181   }
182 
183 
184   train_acc<-mean(y_pred==y)
185 
186   if(mcmc){
```

```
187     Sigma<<-cov(b)
188   }
189
190
191
192   modelBSVM<-list(coef = mb, loss = fvals, MCMCloss = mfval, iter =
        i, lambda = lambda, center = center,
193                     scale = scale, coef_path = b, Sigma = Sigma,
       train_acc = train_acc)
194   class(modelBSVM)<-"LinearBSVM"
195
196
197   return(modelBSVM)
198 }
199
200
201
202
203
204
205 objective<-function(X, y, b, lambda,alpha){
206   f<-lambda^(alpha)*sum(abs(b)^alpha) + sum(ifelse(1-y*X%*%b>0,1-y*
       X%*%b,0))
207   return(f)
208 }
209
210
211
212 est_invnu<-function(X,y,b,mcmc){
213
214   invn<-1/abs(1-y*(X%*%b))
215   indinf<-is.infinite(invn)
216   if(any(indinf)){
217     invn[indinf]<-max(invn[invn!=Inf])^2
218   }
219   if(mcmc){
220     for(j in 1:nrow(X)){
221       invn[j]<-rinvgauss(n = 1,mean = invn[j],shape = 1)
222     }
223   }
224
225   return(invn)
```

```
}



est_invomega<-function(alpha,b,lambda,mcmc,k){

  invo<-alpha*lambda^(alpha-2)*abs(b)^(alpha-2)
  if(mcmc){
    for(l in 1:k){
      invo[l]<-rinvgauss(n = 1,mean = invo[l],shape = 1)
    }
  }

  return(invo)
}



est_lambda<-function(alpha,b,mcmc,a_lambda,b_lambda,k){

  if(mcmc){
    lambda<-rgamma(n = 1, shape = a_lambda+length(b)/alpha, rate =
   b_lambda+sum(abs(b)^alpha))
    lambda<-lambda^(1/alpha)
  }else{
    lambda<-(k/alpha+a_lambda-1)/(b_lambda+sum(abs(b)^alpha))
  }
  lambda<-lambda^(1/alpha)

  return(lambda)
}



est_b<-function(X,y,invnu,invo,lambda,mcmc,k){

  invsigma<-t(X)%*%diag(as.vector(invnu))%*%X + diag(lambda^2,k)%*%
   diag(as.vector(invo))
  Sigma<-solve(invsigma)
  m<-Sigma%*%t(X*y)%*%(1+invnu)

  if(mcmc){
```

```
266     b<-mvrnorm(n = 1,mu = m,Sigma = Sigma)
267   }else{
268     b<-m
269   }
270 
271   return(b)
272 }
273 
274 
275 
276 
277 
278 
279 testBSVM<-function(model,X,y){
280 
281   if(any(names(X)!=names(model$center))){
282     stop("Variables do not match training data.")
283   }
284 
285 
286 
287   if(is.logical(y) | all(y %in% c(0,1))){
288     y<-y*2-1
289   }else if(! all(y %in% c(-1,1))){
290     stop("invalid y - should be binary, logical or +/-1")
291   }
292 
293 
294 
295   X<-scale(X,center = model$center, scale = model$scale)
296   n<-nrow(X)
297   X<-cbind(rep(1,n),X) #additional column for intercept
298 
299 
300   y_pred<-X%*%model$coef>=0
301   y_pred<-y_pred*2-1
302 
303   acc<-mean(y_pred==y)
304   f1<-F1_Score(y,y_pred,positive = '1')
305   cm<-table(y,y_pred)
306 
307 
```

```
308   #p(y=1|X=x)
309   X<-as.matrix(X)
310   y_prob<-c()
311
312   coef0<-which(model$coef==0)
313
314   if(length(coef0)>0){
315     for(i in 1:n){
316       y_prob[i]<-pnorm((X[i,-coef0]%*%model$coef[-coef0])/(t(X[i,-
      coef0])%*%model$Sigma%*%X[i,-coef0]+1))
317     }
318   }else{
319     for(i in 1:n){
320       y_prob[i]<-pnorm((X[i,]%*%model$coef)/(t(X[i,])%*%model$Sigma
      %*%X[i,]+1))
321     }
322   }
323
324
325
326   out<-list(Accuracy = acc, F1_score = f1, Confusion_Matrix = cm,
      p1 = y_prob)
327
328
329   return(out)
330 }
331
332
333
334
335
336 test_dataset_prep<-function(dataset){
337
338   if(! dataset %in% c('spam', 'BreastCancer', 'Ionosphere', '
      PimaIndiansDiabetes', 'Sonar')){
339     stop("Dataset is not included in test.
340          Available: spam, BreastCancer, Ionosphere,
      PimaIndiansDiabetes, Sonar")
341   }
342
343   if(dataset=='spam'){
344     data(spam)
```

```
345     spam<-spam %>% arrange(type)
346
347     y<-ifelse(spam$type=='spam',1,0)
348     X<-spam%>%dplyr::select(-c('type','num3d'))
349
350   }else if(dataset=='BreastCancer'){
351     data(BreastCancer)
352     BreastCancer<-BreastCancer %>% arrange(Class)
353
354     y<-ifelse(BreastCancer$Class=='malignant',1,0)
355     X<-BreastCancer%>%dplyr::select(-c('Id','Class'))
356     X$Bare.nuclei[is.na(X$Bare.nuclei)]<-1
357
358   }else if(dataset=='Ionosphere'){
359     data(Ionosphere)
360     Ionosphere<-Ionosphere %>% arrange(Class)
361
362     y<-ifelse(Ionosphere$Class=='bad',1,0)
363     X<-Ionosphere%>%dplyr::select(-c('V2','Class'))
364
365   }else if(dataset=='PimaIndiansDiabetes'){
366     data(PimaIndiansDiabetes)
367     PimaIndiansDiabetes<-PimaIndiansDiabetes %>% arrange(diabetes)
368
369     y<-ifelse(PimaIndiansDiabetes$diabetes=='pos',1,0)
370     X<-PimaIndiansDiabetes%>%dplyr::select(-'diabetes')
371
372   }else{
373     data(Sonar)
374     Sonar<-Sonar %>% arrange(Class)
375
376     y<-ifelse(Sonar$Class=='R',1,0)
377     X<-Sonar%>%dplyr::select(-'Class')
378   }
379
380   #factor to numeric
381   indx <- sapply(X, is.factor)
382   X[indx] <- lapply(X[indx], function(x) as.numeric(as.character(x)
       ))
383
384   return(list(X,y))
385 }
```

Bibliografija

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[4] J. L. Folks and R. Chhikara. The inverse gaussian distribution and its statistical application—a review. *Journal of the royal statistical society series b-methodological*, 40:263–275, 1978.

[5] M. Goldstein and A. F. M. Smith. Ridge-type Estimators for Regression Analysis. *Journal of the Royal Statistical Society, Series B: Methodological*, 36:284–291, 1974.

[6] E. Gómez-Sánchez-Manzano, M. A. Gómez-Villegas, and J. M. Marín. Multivariate exponential power distributions as mixtures of normal distributions with bayesian applications. *Communications in Statistics - Theory and Methods*, 37(6):972–985, 2008.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[8] D. F. Andrews i C. L. Mallows. Scale Mixtures of Normal Distributions. *Journal of the Royal Statistical Society, Series B: Methodological*, 36:99–102, 1974.

[9] Bent Jørgensen. *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Springer New York, 1982.

[10] Hewlett-Packard Labs. Spam e-mail database. `https://archive.ics.uci.edu/ml/datasets/spambase`.

[11] Chuanhai Liu and Donald B. Rubin. The ecme algorithm: A simple extension of em and ecm with faster monotone convergence. *Biometrika*, 81(4):633–648, 1994.

[12] Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[13] Anđelka Zečević Mladen Nikolić. Mašinsko učenje, 2019.

[14] S. Nadarajah. A generalized normal distribution. *Journal of Applied Statistics*, 32:7:685–694, 2005.

[15] National Institute of Diabetes, Digestive, and Kidney Diseases. Pima indians diabetes database. `https://www.kaggle.com/uciml/pima-indians-diabetes-database`.

[16] University of Wisconsin Hospitals. Breast cancer wisconsin (original) data set. `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)`.

[17] R.D. Pierce. Application of the positive alpha-stable distribution. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, pages 420–424, 1997.

[18] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[19] Nicholas G. Polson and Steven L. Scott. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1 – 23, 2011.

[20] Christian P. Robert. *The Metropolis–Hastings Algorithm*, pages 1–15. American Cancer Society, 2015.

[21] Johns Hopkins University Space Physics Group, Applied Physics Laboratory. Ionosphere data set. `https://archive.ics.uci.edu/ml/datasets/ionosphere`.

[22] R. Paul Gorman Terry Sejnowski. Sonar, mines vs. rocks. `https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)`.

[23] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B: Methodological*, 58:267–288, 1996.

[24] F. Wenzel, Théo Galy-Fajou, M. Deutsch, and M. Kloft. Bayesian nonlinear support vector machines for big data. *ArXiv*, abs/1707.05532, 2017.

[25] Mike West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 09 1987.

Biografija autora

Ana Nikolić rođena je 24.7.1996. u Beogradu. Devetu gimnaziju „Mihailo Petrović Alas" završila je 2015. godine. Osnovne studije upisala je na Matematičkom fakultetu u Beogradu, smer Statistika, aktuarska i finansijska matematika, koje je završila 2019. godine sa prosečnom ocenom 9,34. Od 2019. godine radi u Rajfajzen banci u sektoru za kontrolu rizika.