MCMC metoda i primjene

Bartoš, Elio

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Elio Bartoš

MCMC I PRIMJENE

Diplomski rad

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Elio Bartoš

MCMC I PRIMJENE

Diplomski rad

Voditelj rada:
prof. dr. sc. Bojan Basrak

Zagreb, 2016.

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

Početkom i sredinom 20. stoljeća u statistici su prevladavale metode frekventističke statistike koje su razvili Ronald A. Fisher, Jerzy Neyman i Egon Pearson, a uvelike se koriste i danas. Međutim, nakon 1950. godine javlja se i grupa tzv. bayesovskih statističara koja zastupa drugačiji pristup u statistici. Zasniva se na Bayesovom teoremu za koji su zaslužni Thomas Bayes i Pierre-Simon Laplace koji su živjeli u 18. stoljeću. Osnovna je ideja bayesovske statistike parametrima od interesa pridružiti vjerojatnosnu razdiobu koja predstavlja naše znanje o tome parametru, a zatim na temelju podataka i naših uvjerenja izraziti novo znanje o parametrima u obliku nove vjerojatnosne razdiobe. Tu dobivenu vjerojatnosnu razdiobu zovemo aposteriori razdioba ili distribucija. Problem kod bayesovske statistike u to vrijeme bio je što je vrlo teško bilo izračunati aposteriori razdiobu. Da bi se točno izračunala, bilo je potrebno izračunati višedimenzionalni integral vrlo komplicirane funkcije. Umjesto da se egzaktno izračuna aposteriori distribucija, ovaj problem rješava se uzimanjem velikog uzorka iz aposteriori distribucije, a to je omogućeno algoritmima Monte Carlo Markovljevih lanaca (eng. *Markov Chain Monte Carlo*), u daljnjem tekstu MCMC.

Općenito, MCMC algoritmi služe za simuliranje uzorka iz komplicirane distribucije koju znamo samo do na konstantu. Njihovo razvijanje i istraživanje krajem 20. stoljeća, uz razvoj računala, dovelo je do sve veće primjene bayesovske statistike jer je omogućilo primjenu na složenije modele koji su mogli bolje opisati podatke iz realnog svijeta. Jedan od prvih algoritama tog tipa bio je Metropolis algoritam iz 1953. godine koji je dobio ime po Nicholasu Metropolisu. Njegovu generalizaciju dao je W. K. Hastings 1970. godine, a algoritam je dobio ime Metropolis-Hastings. To je glavni algoritam koji ćemo obraditi u ovome radu. MCMC algoritmi široko su primjenjivi, a osim u bayesovskoj statistici koriste se i u računalnoj fizici, računalnoj biologiji i računalnoj lingvistici. Kako bismo precizno definirali MCMC algoritme, navodimo sljedeću definiciju čiji će potpun smisao biti jasan nakon čitanja prvog poglavlja.

Definicija 0.1. *MCMC algoritam za simuliranje uzorka iz distribucije dane gustoćom f je svaka metoda koja generira ergodski Markovljev lanac (X_n) čija je stacionarna distribucija f.*

Iako MCMC algoritmi mogu biti vrlo jednostavni za implementaciju, u njihovoj pozadini je teorija Markovljevih lanaca na općenitom skupu stanja. Zato u prvom poglavlju ovog rada uvodimo pojmove iz te teorije, te navodimo neke bitne rezultate koji će nam biti potrebni kako bismo dokazali svojstva Metropolis-Hastings algoritma.

U drugom poglavlju bavimo se Metropolis-Hastings algoritmom. Objašnjavamo algoritam i uz pomoć pojmova iz prvog poglavlja dokazujemo njegova svojstva i valjanost. Zatim prelazimo na neke njegove varijante kao što su Nezavisni Metropolis-Hastings algoritam i Metropolis-Hastings algoritam sa slučajnom šetnjom te na primjeru ilustriramo njihovo korištenje. Zatim se kratko dotičemo i Gibbsovog uzorkovanja koji je jedan od najvažnijih MCMC algoritama.

U trećem poglavlju objašnjavamo osnovne ideje bayesovske statistike kako bismo objasnili važnost primjene MCMC algoritama u bayesovskoj statistici. Zatim ukratko

objašnjavamo generalizirane linearne modele kako bismo mogli napraviti kompleksan model iz perspektive bayesovske statistike. Napravili smo model linearne regresije s teškim repovima i pomoću MCMC algoritama dobili rezultate. Dotičemo se i raznih tema kojima se treba posvetiti prilikom korištenja bayesovske statistike i MCMC algoritama. U dodatku opisujemo tehničke detalje o tome kako smo izveli primjer u programskom jeziku R i programu JAGS.

Poglavlje 1

Markovljevi lanci na općenitom skupu stanja

Kako bismo precizno razumjeli MCMC algoritme, u ovom poglavlju ukratko ćemo objasniti Markovljeve lance na općenitom skupu stanja. Skup stanja označimo sa $\mathcal{X}$, a sa $\mathcal{B}(\mathcal{X})$ označimo σ-algebru na $\mathcal{X}$. Iako većina rezultata u ovom poglavlju vrijedi općenito, mi ćemo pretpostavljati da je $\mathcal{X} = \mathbb{R}^m$ za neki $m \in \mathbb{N}$, a $\mathcal{B}(\mathcal{X})$ je Borelova σ-algebra $\mathcal{B}(\mathbb{R}^m)$. Prisjetimo se da je $\mathcal{B}(\mathbb{R}^m)$ σ-algebra generirana svim otvorenim skupovima u $\mathbb{R}^m$. Intuitivno Markovljev lanac možemo zamišljati kao niz slučajnih skokova iz jednog stanja u drugo pri čemu skok u stanje ne ovisi o tome gdje smo prije bili, već samo o trenutnom stanju u kojem se nalazimo. Dakle, za svako stanje u kojem se nalazimo, trebamo znati vjerojatnosnu distribuciju idućeg skoka. U skladu s time imamo sljedeću definiciju.

Definicija 1.2. Prijelazna jezgra *(eng. transition kernel) K je funkcija definirana na $\mathcal{X} \times \mathscr{B}(\mathcal{X})$ takva da vrijedi*

(i) za svaki $x \in \mathcal{X}$, $\quad K(x, \cdot)$ je vjerojatnosna mjera,

(ii) za svaki $A \in \mathscr{B}(\mathcal{X})$, $\quad K(\cdot, A)$ je izmjeriva funkcija na $\mathcal{X}$.

Ako je za svaki $x \in \mathcal{X}$ mjera $K(x, \cdot)$ apsolutno neprekidna u odnosu na Lebesgueovu mjeru $\lambda^{(m)}$, onda ćemo sa $k(x, \cdot) : \mathcal{X} \to [0, \infty\rangle$ označavati uvjetnu gustoću s obzirom na mjeru $K(x, \cdot)$, tj.

$$K(x, A) = \int_A k(x, y)\, \lambda^{(m)}(dy).$$

Znamo da takva funkcija postoji prema Radon-Nikodymovom teoremu za mjere. Ovaj slučaj ćemo zvati neprekidan slučaj. Sada definiramo Markovljev lanac na skupu stanja $\mathcal{X}$.

Definicija 1.3. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Slučajni proces $X = (X_n : n \geq 0) : \Omega \to \prod_{i=0}^{\infty} X_i$ izmjeriv u odnosu na $(\mathcal{F}, \prod_{i=0}^{\infty} \mathcal{B}(\mathcal{X}_i))$ zovemo vremenski homogen* **Markovljev lanac** *s prijelaznom jezgrom K i početnom distribucijom μ ako za svaki $n \in \mathbb{N}_0$ vrijedi*

$$\begin{aligned} &\mathbb{P}(X_0 \in A_0, X_1 \in A_1, \ldots, X_n \in A_n) = \\ &\qquad \int_{A_0} \ldots \int_{A_{n-1}} K(y_{n-1}, A_n) K(y_{n-2}, dy_{n-1}) \ldots K(y_0, dy_1) \mu(dy_0), \end{aligned} \tag{1}$$

za sve $A_0, A_1, \ldots, A_n \subseteq \mathcal{X}$.

Može se pokazati da za svaku prijelaznu jezrgu K i početnu distribuciju μ postoji vjerojatnosni prostor i na njemu slučajan proces tako da vrijedi (1). Detaljnije o tome može se naći u Meyn i Tweedie [5] i u Lisičar [4].

Izraz $\prod_{i=0}^{\infty} \mathcal{B}(\mathcal{X}_i)$ zovemo produkt σ-algebri $\mathcal{B}(\mathcal{X}_i)$ pri čemu je $\mathcal{X}_i = \mathcal{X}$ za $i \in \mathbb{N}_0$. Njezinu definiciju može se pronaći u Sarapa [7] (str. 356).

Ukoliko je $X : \Omega \to \mathcal{X}$ izmjeriva funkcija u odnosu na $(\mathcal{F}, \mathcal{B}(\mathcal{X}))$ te μ vjerojatnosna mjera na $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, tada ćemo sa $X \sim \mu$ označavati da je zakon razdiobe od X jednak μ, tj. da vrijedi

$$\mathbb{P}(X \in A) = \mathbb{P}_X(A) = \mu(A),$$

za svaki $A \in \mathcal{F}$.

Iz definicije vidimo da važnu ulogu pri konstruiranju Markovljevog lanca osim prijelazne jezgre igra i početna mjera (početna distribucija). Ukoliko je $X_0 \sim \mu$, gdje je μ vjerojatnosna mjera na $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, tada kažemo da je μ početna distribucija lanca, a vjerojatnost $\mathbb{P}$ iz definicije 1.3 označavat ćemo sa $\mathbb{P}_\mu$. Ukoliko lanac kreće iz stanja $x_0 \in \mathcal{X}$, što odgovara mjeri koncentriranoj u x_0 (oznaka mjere δ_{x_0}), onda ćemo pisati $\mathbb{P}_{x_0}$ umjesto $\mathbb{P}_{\delta_{x_0}}$.

Kada bismo riječima išli opisati Markovljev lanac, rekli bismo da je distribucija budućeg stanja X_{n+1} uz dano sadašnje stanje x_n i prijašnja stanja $x_{n-1}, \ldots, x_0$ ista kao i distribucija od X_{n+1} uz dano sadašnje stanje x_n. Kada kažemo 'uz dano stanje x_i', mislimo na to da u integralu (1) umjesto $K(y_i, \cdot)$ koristimo mjeru $K(x_i, \cdot)$. Razlika je u tome što integriramo po varijabli y_i, a mi ju zamijenimo s konkretnom vrijednosti x_i. Analogno ako je zadano više stanja.

Vremenska homogenost znači da je distribucija od $(X_{n_1}, X_{n_2}, \ldots, X_{n_k})$ uz dano x_{n_0} ista kao i distribucija od $(X_{n_1-n_0}, X_{n_2-n_0}, \ldots, X_{n_k-n_0})$ uz dano x_0 za svaki $k \in \mathbb{N}$ i $n_0 \leq n_1 \leq \ldots \leq n_k$ prirodne brojeve. Vremenska homogenost znači da se lanac jednako ponaša kroz vrijeme. Na primjer, ako znamo da se nalazimo u stanju x, za buduće kretanje nam nije bitno nalazimo li se u trenutku $n = 0$ u tom stanju ili u trenutku $n = 100$; buduće kretanje lanca je jednako. To je jasno i iz toga što je vjerojatnosna mjera budućeg skoka u tom slučaju dana sa $K(x, \cdot)$, a vidimo da to ne ovisi o n. Kako ćemo razmatrati samo vremenski homogene Markovljeve lance, izraz vremenski homogen ćemo izostavljati.

Primjer 1.4. AR(1) proces
Jednostavan primjer Markovljevog lanca na $\mathbb{R}$ možemo definirati na sljedeći način:

$$X_n = \theta X_{n-1} + \epsilon_n\,, \quad \theta \in \mathbb{R},$$

pri čemu su $\epsilon_n \sim N(0, \sigma^2)$ međusobno nezavisne za svaki $n \in \mathbb{N}$. Za početno stanje uzmimo na primjer $X_0 = 0$. Početna distribucija je $\mu = \delta_0$. Jasno je da ako se nalazimo u stanju x, distribucija idućeg koraka je $N(\theta x, \sigma^2)$, čime smo dobili prijelaznu jezgru K. Dokaz da je $AR(1)$ uistinu Markovljev lanac, može se naći u Lisičar [4].

Promotrimo još malo jednadžbu (1) koja nam govori kako računamo konačno dimenzionalne distribucije Markovljevog lanca, u specijalnom slučaju. Pogledajmo za

neki $x \in \mathcal{X}$ i $A \in \mathcal{B}(\mathcal{X})$ izraze

$$\begin{aligned}
\mathbb{P}_x(X_1 \in A) &= K(x, A), \\
\mathbb{P}_x(X_2 \in A) &= \int_{\mathcal{X}} K(y, A)\, K(x, dy), \\
\mathbb{P}_x(X_n \in A) &= \underbrace{\int_{\mathcal{X}} \dots \int_{\mathcal{X}}}_{n-1 \text{ puta}} K(y_{n-1}, A)\, K(x, dy_1) K(y_1, dy_2) \dots K(y_{n-2}, dy_{n-1}).
\end{aligned}$$

Prokomentirajmo drugu jednakost. Primijetimo da je zbog zahtjeva *(ii)* u definiciji prijelazne jezgre 1.2 $K(\cdot, A)$ izmjeriva funkcija, tj. slučajna varijabla. Integriramo po mjeri $K(x, \cdot)$. Intuitivno zamišljamo da ako u drugom koraku želimo biti u skupu A, tada u prvom koraku moramo skočiti bilo gdje (zato je integral po cijelom skupu $\mathcal{X}$) i zatim iz tog stanja skočiti u skup A.

Sada možemo definirati prijelaznu jezgru u n koraka. Prvo označimo $K^1(x, A) = K(x, A)$. Definiramo za $n > 1$:

$$K^n(x, A) = \int_{\mathcal{X}} K^{n-1}(y, A)\, K(x, dy).$$

Odmah vidimo da je $\mathbb{P}_x(X_n \in A) = K^n(x, A)$. Sada smo spremni definirati ireducibilnost Markovljevog lanca.

Definicija 1.5. *Neka je dana mjera ϕ na $\mathcal{B}(\mathcal{X})$. Markovljev lanac (X_n) s prijelaznom jezgrom K je* **ϕ-ireducibilan** *ako*

$$\forall x \in \mathcal{X} \text{ i } A \in \mathcal{B}(\mathcal{X}) \text{ t.d. } \phi(A) > 0 \implies \exists n \in \mathbb{N} \text{ t.d. } K^n(x, A) > 0.$$

Lanac je **jako ϕ-ireducibilan** *ukoliko je $n = 1$ za svaki $x \in \mathcal{X}$ i svaki $A \in \mathcal{B}(\mathcal{X})$.*

Ireducibilnost u odnosu na mjeru ϕ znači da za svaki skup koji nije mjere nula postoji pozitivna vjerojatnost dolaska u taj skup u konačno mnogo koraka. Ireducibilnost je bitna jer govori da lanac nije osjetljiv na početne uvjete jer ima pozitivnu vjerojatnost da posjeti sve skupove koji nisu mjere nula, u odnosu na ϕ, bez obzira iz kojeg stanja lanac krenuo. To će nam biti bitno kod MCMC algoritama jer nećemo morati brinuti o pogodnim početnim pozicijama iz kojeg će naš lanac krenuti.

U teoriji je zanimljivo promatrati i broj dolazaka u skup A (broj prolazaka kroz skup A). Zato definiramo

$$\eta_A = \sum_{n=1}^{\infty} \mathbb{1}_A(X_n).$$

Za ireducibilne lance znamo da imaju pozitivnu vjerojatnost dolaska u izmjerive skupove ne-nul mjere. Sada nas zanima koliki je očekivani broj dolazaka u te skupove. U skladu s time imamo sljedeću definiciju.

Definicija 1.6. *Markovljev lanac (X_n) je* **povratan** *ako*

(i) postoji mjera ψ takva da je (X_n) ψ-ireducibilan i

(ii) za svaki skup $A \in \mathcal{B}(\mathcal{X})$ takav da je $\psi(A) > 0$ vrijedi $\mathbb{E}_x[\eta_A] = \infty$ za svaki $x \in A$.

Za povratne lance vrijedi da ukoliko kreću iz izmjerivog skupa A ne-nul mjere, tada je očekivani broj povrataka u to stanje beskonačan. Postoji i jači oblik povratnosti koji zahtjeva čak i to da za svaku putanju lanca beskonačno puta posjetimo skup A. Tu vrstu povratnosti uveo je Harris 1956. godine.

Definicija 1.7. *Markovljev lanac (X_n) je* **Harris povratan** *ako*

(i) postoji mjera ψ takva da je (X_n) ψ-ireducibilan i

(ii) za svaki izmjerivi skup A takav da je $\psi(A) > 0$ vrijedi $\mathbb{P}_x(\eta_A = \infty) = 1$ za sve $x \in A$.

Primijetimo da zbog ψ-ireducibilnosti u prethodnoj definiciji znamo da ćemo za svaki $x \in \mathcal{X}$ doći u skup A u konačno mnogo koraka, a nakon toga, zbog toga što se radi o Markovljevom lancu i zbog drugog svojstva u definiciji 1.7, posjetit ćemo skup A beskonačno mnogo puta. Dakle, svaki skup ne-nul mjere posjetit ćemo beskonačno mnogo puta bez obzira iz kojeg stanja $x \in \mathcal{X}$ krenuli.

U teoriji Markovljevih lanaca na diskretnom skupu stanja D definira se pojam aperiodičnosti. Za stanje $i \in D$ period stanja i definira se kao najveći zajednički djelitelj skupa

$$\{n \geq 1 : \mathbb{P}_i(X_n = i) > 0\}.$$

Intuitivno, ako je period stanja iz kojeg krećemo 2, tada nećemo moći biti u tom stanju u neparnim koracima. Pokazuje se da je kod ireducibilnih lanaca period jednak za svako stanje. Ukoliko on iznosi 1, tada lanac zovemo aperiodičnim. Jasno je da je dovoljan uvjet da ireducibilan lanac bude aperiodičan taj da je $\mathbb{P}_i(X_1 = i) > 0$, tj. da lanac može ostati u istom stanju. Za lance na općenitom skupu stanja $\mathcal{X}$ komplicira se definicija ovog pojma pa ga nećemo definirati. Zainteresiranog čitatelja upućujemo na Robert i Castella [6]. Umjesto definicije dat ćemo dovoljan uvjet koji vrijedi i u diskretnom slučaju.

Napomena 1.8. Ukoliko je Markovljev lanac ϕ-ireducibilan i vjerojatnost da ostane u istom stanju strogo je veća od 0, onda je lanac **aperiodičan**.

Pod marginalnom distribucijom lanca (X_n) mislimo na distribuciju od X_n za $n \in \mathbb{N}$. Sada nam je cilj uvesti definiciju koje opisuje svojstvo da marginalne distribucije lanca ne ovise o vremenu n. Pretpostavimo da znamo $X_n \sim \pi$ pri čemu je π vjerojatnosna mjera. Tada je

$$\mathbb{P}_\mu(X_{n+1} \in A) = \mathbb{P}_\mu(X_{n+1} \in A, X_n \in \mathcal{X}) = \int_{\mathcal{X}} K(x, A)\, \pi(dx).$$

Definicija 1.9. *Za σ-konačnu mjeru π kažemo da je* **invarijantna mjera** *za prijelaznu jezgru K (i pridruženi lanac) ako vrijedi*

$$\pi(A) = \int_{\mathcal{X}} K(x, A)\, \pi(dx), \quad \text{za svaki } A \in \mathcal{B}(\mathcal{X}).$$

Definicija nam kaže ukoliko $X_n \sim \pi$, onda je i $X_{n+1} \sim \pi$, tj. neovisno o vremenu marginalna distribucija lanca ostala je ista. Ukoliko je π vjerojatnosna mjera, invarijantnu mjeru često nazivamo **stacionarna mjera** ili distribucija.

Uočimo da ukoliko je π stacionarna distribucija lanca i uspijemo postići da je u nekom trenutku $X_n \sim \pi$, tada svaka iduća vrijednost lanca dolazi iz π distribucije što nam omogućuje da uzorkujemo iz te distribucije, iako uzorak neće nužno biti nezavisan. Jasno je da želimo da ovo svojstvo ima naš lanac konstruiran MCMC algoritmom. Ponekad će biti teško provjeriti direktno to svojstvo, no zato u nastavku definiramo dovoljan uvjet da bi lanac imao invarijantnu mjeru.

Definicija 1.10. *Neka je K prijelazna jezgra u neprekidnom slučaju, s uvjetnom gustoćom k. Markovljev lanac s prijelaznom jezgrom K zadovoljava* **uvjet detaljne ravnoteže** *ukoliko postoji funkcija $f : \mathcal{X} \to \mathbb{R}$ takva da vrijedi*

$$k(y, x)f(y) = k(x, y)f(x), \quad \text{za sve } (x, y) \in \mathcal{X} \times \mathcal{X}.$$

Ovaj uvjet nije nužan da f bude stacionarna distribucija lanca s jezgrom prijelaza K, no daje dovoljne uvjete koje nije teško provjeriti. Upravo preko ovoga uvjeta dokazat ćemo da lanac konstruiran MCMC algoritmima ima stacionarnu distribuciju f. U idućem teoremu dokazujemo da je uvjet detaljne ravnoteže uistinu dovoljan za postojanje stacionarne distribucije.

Teorem 1.11. *Pretpostavimo da Markovljev lanac s prijelaznom jezgrom K zadovoljava uvjet detaljne ravnoteže s vjerojatnosnom funkcijom gustoće π. Tada je gustoća π invarijantna gustoća lanca.*

Dokaz. Napomenimo da kada kažemo da je gustoća invarijantna za Markovljev lanac, mislimo na mjeru ν generiranu tom gustoćom, tj.

$$\nu(B) = \int_B \pi(x)\, \lambda^{(m)}(dx), \quad B \in \mathcal{B}(\mathcal{X}).$$

Za izmjerivi skup B računamo

$$\begin{aligned}
&\int_{\mathcal{X}} K(y, B)\, \nu(dy) = \int_{\mathcal{X}} K(y, B)\pi(y)\, \lambda^{(m)}(dy) \\
&= \int_{\mathcal{X}} \int_B k(y, x)\pi(y)\, \lambda^{(m)}(dx)\lambda^{(m)}(dy) = \int_{\mathcal{X}} \int_B k(x, y)\pi(x)\, \lambda^{(m)}(dx)\lambda^{(m)}(dy) \\
&= \int_B \pi(x) \Big(\underbrace{\int_{\mathcal{X}} k(x, y)\, \lambda^{(m)}(dy)}_{K(x,\mathcal{X})=1} \Big)\, \lambda^{(m)}(dx) = \int_B \pi(x)\, \lambda^{(m)}(dx) = \nu(B),
\end{aligned}$$

pri čemu smo iskoristili uvjet detaljne ravnoteže u trećoj jednakosti te Fubinijev teorem u četvrtoj jednakosti, što je opravdano jer su podintegralne funkcije nenegativne. □

U praksi nije lako postići da $X_n \sim \pi$ za fiksni n, pri čemu je π stacionarna distribucija lanca. Zato nas zanima granična distribucija od X_n , tj. prema čemu lanac konvergira. U nastavku iskazujemo teorem koji daje dovoljne uvjete na lanac (X_n) kako bi njegova granična distribucija bila π, čime dobivamo da za velike n realizacije lanca možemo smatrati dobivenim iz distribucije π.

Norma u kojoj ćemo gledati konvergenciju je totalna varijacijska norma definirana izrazom

$$\|\mu_1 - \mu_2\|_{TV} = \sup_{A \in \mathcal{B}(\mathcal{X})} |\mu_1(A) - \mu_2(A)|.$$

Teorem 1.12. *Ako je Markovljev lanac (X_n) Harris povratan i aperiodičan sa stacionarnom mjerom π, onda vrijedi*

$$\lim_{n\to\infty} \left\| \int_{\mathcal{X}} K^n(x, \cdot)\, \mu(dx) - \pi \right\|_{TV} = 0,$$

za svaku početnu distribuciju μ.

Raspišimo tvrdnju teorema. Prema definiciji totalne varijacijske norme imamo

$$\sup_{A \in \mathcal{B}(\mathcal{X})} \left| \int_{\mathcal{X}} K^n(x, A)\, \mu(dx) - \pi(A) \right| \longrightarrow 0, \quad n \to \infty,$$

iz čega slijedi

$$\forall A \in \mathcal{B}(\mathcal{X}), \quad \int_{\mathcal{X}} K^n(x, A)\, \mu(dx) \longrightarrow \pi(A), \quad n \to \infty,$$

tj.

$$\forall A \in \mathcal{B}(\mathcal{X}), \quad \mathbb{P}_\mu(X_n \in A) \longrightarrow \pi(A), \quad n \to \infty.$$

Dakle, ovaj teorem govori nam da marginalne distribucije lanca, neovisno o početnoj distribuciji ili točki iz koje naš lanac kreće, konvergiraju prema stacionarnoj distribuciji. Za velike n marginalna distribucija od X_n aproksimira stacionarnu distribuciju π.

Kako bismo realizaciju lanca mogli koristiti kao uzorak dobiven iz distribucije π, potreban nam je Ergodski teorem. Uz dane opservacije $X_1, X_2, \ldots, X_n$ Markovljevog lanca, zanimat će nas ponašanje parcijalnih suma

$$S_n(h) = \frac{1}{n} \sum_{i=0}^{n} h(X_i),$$

pri čemu je h integrabilna funkcija s obzirom na odgovarajući prostor mjere.

Teorem 1.13 (Ergodski teorem)**.** *Ako Markovljev lanac (X_n) ima σ-konačnu invarijantnu mjeru π, ekvivalentno je:*

(i) ako su $f, g \in L^1(\pi)$ takve da $\int_{\mathcal{X}} g(x)\,\pi(dx) \neq 0$, tada

$$\lim_{n\to\infty} \frac{S_n(f)}{S_n(g)} = \frac{\int_{\mathcal{X}} f(x)\,\pi(dx)}{\int_{\mathcal{X}} g(x)\,\pi(dx)},$$

(ii) Markovljev lanac (X_n) je Harris povratan.

Ergodski teorem koristit ćemo tako da dokažemo pretpostavke i tvrdnju *(ii)* te će tada vrijediti tvrdnja *(i)*. Markovljev lanac koji je aperiodičan i Harris povratan, zovemo **ergodski** Markovljev lanac. Ukoliko ergodski Markovljev lanac ima stacionarnu distribuciju π, onda na njega možemo primijeniti ergodski teorem 1.13 i teorem 1.12, a to će nam biti ključno kod dokazivanja valjanosti MCMC algoritama.

Uočimo da ukoliko je π vjerojatnosna mjera, ona je σ-konačna. Također, ukoliko uzmemo $g = 1$, imamo $\int_{\mathcal{X}} g(x)\,\pi(dx) = 1$ pa slijedi

$$\lim_{n\to\infty} S_n(f) = \int_{\mathcal{X}} f(x)\,\pi(dx) = \mathbb{E}_\pi[f(X)].$$

Oznaka π kod očekivanja znači da je slučajna varijabla X distribuirana prema mjeri π, tj. zakon razdiobe od X je $\mathbb{P}_X = \pi$. Ovaj teorem nam omogućuje da realizaciju lanca koristimo za izračunavanje raznih statistika vezanih uz distribuciju π, iako realizacija lanca ne daje nezavisan uzorak. Zbog toga je on ključan za primjenu MCMC algoritama.

Teorem 1.12 govori nam o asimptotskoj konvergenciji marginalnih distribucija lanca prema stacionarnoj distribuciji, ali ne govori ništa o brzini te konvergencije. Kako bismo opisali brzinu konvergencije, uvodimo sljedeća svojstva.

Definicija 1.14. *Ergodski Markovljev lanac sa stacionarnom distribucijom π je* **uniformno ergodski** *Markovljev lanac ako postoje $\rho < 1$ i $M < \infty$ takvi da za svaki $x \in \mathcal{X}$ vrijedi*

$$\|K^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq M\rho^n,$$

za svaki $n \in \mathbb{N}$.

Iz definicije odmah slijedi

$$\lim_{n\to\infty} \sup_{x\in\mathcal{X}} \|K^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq \lim_{n\to\infty} M\rho^n = 0.$$

Slabije svojstvo od uniformne ergodičnosti lanca je geometrijska ergodičnost lanca.

Definicija 1.15. *Ergodski Markovljev lanac sa stacionarnom distribucijom π je* **geometrijski ergodski** *Markovljev lanac ako postoji $\rho < 1$ takav da*

$$\|K^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq M(x)\rho^n,$$

za svaki $n \in \mathbb{N}$, pri čemu je $M(x) < \infty$ za π gotovo sve $x \in \mathcal{X}$.

Razlika je u tome što sada konstanta M ovisi o početnom stanju x. Ovo svojstvo govori nam da $\|K^n(x,\cdot) - \pi(\cdot)\|_{TV}$ opada barem geometrijski brzo. Ova će nam svojstva biti bitna jer pružaju dovoljne uvjete za primjenu centralnog graničnog teorema za Markovljeve lance.

Poglavlje 2

Metropolis-Hastings algoritam

2.1 Uvod

U ovom poglavlju objasnit ćemo najopćenitiji algoritam za simuliranje uzorka iz distribucije zadane vjerojatnosnom funkcijom gustoće f pomoću Markovljevih lanaca - Metropolis-Hastings algoritam. Možda zvuči neobično krenuti najopćenitijim algoritmom, ali ovaj je algoritam zapravo najjednostavniji i za objasniti i za direktnu implementaciju, a zbog svoje općenitosti nudi mnogo mogućnosti. Prisjetimo se, problem je simulirati uzorak iz distribucije zadane gustoćom f koju znamo do na normalizirajuću konstantu kako bismo mogli aproksimirati integrale oblika

$$I = \int_X h(x)f(x)\lambda^{(m)}(dy) = \mathbb{E}_f[h(X)]. \tag{2}$$

Kako bismo to napravili, konstruirat ćemo ergodski Markovljev lanac čija je stacionarna distribucija f. Da bismo konstruirali takav Markovljev lanac, trebamo konstruirati prijelaznu jezgru kojoj je f stacionarna distribucija i osigurati konvergenciju marginalnih distribucija lanca (X_n) prema f te osigurati da možemo primijeniti Ergodski teorem 1.13. Na kraju ćemo dobiti aproksimaciju integrala (2) oblika

$$\hat{I}_N = \frac{1}{N}\sum_{n=1}^{N} h(X_n). \tag{3}$$

2.2 Algoritam

Distribuciju f iz koje želimo simulirati uzorak, zvat ćemo ciljna (eng. *target*) distribucija. Kako bismo konstruirali Markovljev lanac, za svako stanje $x \in \mathcal{X}$ u kojem se lanac može naći, moramo definirati vjerojatnosnu mjeru $K(x, \cdot)$. Nju ćemo zadati preko funkcije gustoće. Dakle, za svaki x zadajemo uvjetnu funkciju gustoće $q(y|x)$ koju zovemo prijedložna distribucija ili gustoća (eng. *proposal*). Tada vrijedi

$$K(x, A) = \int_A q(y|x)\,\lambda^{(m)}(dy), \quad A \in \mathcal{B}(\mathcal{X}).$$

Na primjer, za $q(y|x)$ možemo uzeti funkciju gustoće normalne slučajne varijable s očekivanjem x i standardnom devijacijom σ. Za algoritam je nužno da znamo simulirati iz tih distribucija te da znamo izračunati omjer

$$\frac{f(y)}{q(y|x)}$$

do na konstantu. Algoritam ide ovako:

Algoritam 1 Metropolis-Hastings algoritam

1: Uz dano $X_n = x_n$
2: Generiraj $Y_n \sim q(\cdot|x_n)$
3: Generiraj $U \sim U([0,1])$
4: Izračunaj $\rho(x_n, Y_n)$, pri čemu je

$$\rho(x,y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1\right\}$$

5: Definiraj

$$X_{n+1} = \begin{cases} Y_n & \text{ako } U \leq \rho(x_n, Y_n) \\ x_n & \text{inače} \end{cases}$$

6: Ponavljaj za $n = n + 1$

Algoritam kreće iz stanja x_0 koje biramo proizvoljno, ali tako da vrijedi $f(x_0) > 0$. Zatim generiramo vrijednost iz *proposal* distribucije. Izraz $\rho(x, y)$ je vjerojatnost da ćemo prihvatiti vrijednost iz *proposal* distribucije i skočiti u to stanje. Ukoliko ne prihvatimo novo stanje, ostajemo u trenutnom stanju i u idućem koraku. Algoritam zaustavljamo kada smatramo da je lanac napravio dovoljno koraka za naše potrebe. O tome ćemo više govoriti kasnije. Uočimo da je zbog omjera $\frac{f(y)}{f(x)}$ gustoću f dovoljno znati samo do na konstantu. Također, zbog omjera $\frac{q(x|y)}{q(y|x)}$ i $q(\cdot|x)$ je potrebno znati samo do na konstantu koja ne ovisi o x. Ukoliko bi ona ovisila o x, imali bismo različite konstante pa se one ne bi skratile u omjeru. Ipak, uvjetne gustoće sami zadajemo pa njih najčešće u potpunosti znamo, no to što nam u algoritmu treba samo njihov omjer, može biti prednost pri implementaciji. Primijetimo da ukoliko krenemo iz stanja x_0 tako da vrijedi $f(x_0) > 0$, onda nam se nikad neće dogoditi da je $f(x_n) = 0$ i da $\rho(x_n, y_n)$ nije definiran jer će vjerojatnost skoka u x_n u koraku prije biti 0.

2.3 Svojstva Metropolis-Hastings algoritma

U ovom dijelu cilj nam je za Markovljev lanac konstruiran algoritmom 1 pokazati da vrijedi:

- f je njegova stacionarna distribucija,
- lanac konvergira prema stacionarnoj distribuciji,
- izraz (3) je aproksimacija integrala (2).

Iako ovaj algoritam daje veliku slobodu pri izboru uvjetne gustoće q, ipak neki minimalni uvjeti moraju biti zadovoljeni kako bi f bila stacionarna distribucija konstruiranog lanca. Ukoliko nosač gustoće f, $\operatorname{supp} f = \{x \in X : f(x) > 0\}$, nije povezan, bitno je da q omogućuje povezivanje svih komponenti, tj. omogućuje lancu skok iz

jedne komponente u drugu. Ne smije se dogoditi da postoji skup $A \subset \operatorname{supp} f$ takav da

$$\int_A f(x)\,\lambda^{(m)}(dx) > 0 \quad \text{i} \quad \int_A q(y|x)\,\lambda^{(m)}(dy) = 0, \quad \text{za svaki} \quad x \in \operatorname{supp} f$$

jer tada lanac nikada neće posjetiti skup A, a uzorak iz f može sadržavati vrijednost iz skupa A. Dovoljan uvjet je da nosač od q sadrži nosač od f tj.

$$\bigcup_{x \in \operatorname{supp} f} \operatorname{supp} q(\cdot|x) \supset \operatorname{supp} f.$$

Uz pomoć tog uvjeta u sljedećem teoremu dokazujemo da je f stacionarna distribucija konstruiranog lanca.

Teorem 2.16. *Neka je (X_n) lanac dobiven Metropolis-Hastings algoritmom 1. Ako nosač od q sadrži nosač od f, tada vrijedi:*

(i) prijelazna jezgra K lanca (X_n) zadovoljava uvjet detaljne ravnoteže sa f,

(ii) f je stacionarna distribucija lanca.

Dokaz. Za dokaz tvrdnje *(i)* trebamo pokazati

$$k(y,x)f(y) = k(x,y)f(x), \quad \text{za sve} \quad x, y \in \mathcal{X}.$$

Prisjetimo se da je $k(x,y)$ uvjetna gustoća prijelaza $K(x,\cdot)$ ukoliko se radi o neprekidnom slučaju. Za proizvoljan $A \in \mathcal{B}(\mathcal{X})$ i $x \in \mathcal{X}$ računamo

$$\begin{aligned}
&K(x,A) = \mathbb{P}_x(X_1 \in A) = \\
&\mathbb{P}_x(X_1 \in A, \text{prihvatimo } Y) + \mathbb{P}_x(X_1 \in A, \text{odbijemo } Y) = \\
&\mathbb{E}[\rho(x,Y)\mathbb{1}_{\{Y \in A\}}] + \mathbb{1}_A(x)\,\mathbb{P}_x(\text{odbijemo } Y) = \\
&\int_A \rho(x,y)q(y|x)\,\lambda^{(m)}(dy) + \underbrace{\delta_x(A)}_{\int_A \delta_x}\Big(1 - \underbrace{\int_A \rho(x,y)q(y|x)\,\lambda^{(m)}(dy)}_{r(x)}\Big),
\end{aligned}$$

pri čemu je $Y \sim q(\cdot|x)$ ponuđena vrijednost, δ_x Diracova mjera koncentrirana u točki x, a mjeru skupa smo zamijenili integralom jedinične funkcije po tom skupu. Dakle, imamo

$$\begin{aligned}
K(x,A) &= \int_A \rho(x,y)q(y|x)\,\lambda^{(m)}(dy) + \int_A (1 - r(x))\delta_x(dy) = \\
&\int_A \rho(x,y)q(y|x)\,\lambda^{(m)}(dy) + \int_A (1 - r(x))\delta_x(y)\,\lambda^{(m)}(dy),
\end{aligned}$$

pri čemu je sada δ_x Diracova delta "funkcija" koju shvaćamo kao:

$$\delta_x(y) = \begin{cases} +\infty & \text{ako } y = x \\ 0 & \text{ako } y \neq x \end{cases} \quad \text{i vrijedi} \quad \int_{\mathcal{X}} \delta_x(y)\,\lambda^{(m)}(dy) = 1.$$

Ovo treba shvatiti samo kao notaciju jer Diracova mjera nije apsolutno neprekidna u odnosu na Lebesgueovu mjeru pa ne postoji funkcija koja bi zadovoljavala gornju jednakost kod integrala.

Na kraju dobivamo uvjetnu gustoću prijelaza:

$$k(x,y) = \rho(x,y)q(y|x) + (1 - r(x))\delta_x(y).$$

Provjerimo sada uvjet detaljne ravnoteže:

$$k(x,y)f(x) = \rho(x,y)q(y|x)f(x) + (1 - r(x))\delta_x(y)f(x).$$

Uočimo da vrijedi $\rho(x,y) < 1 \implies \rho(y,x) = 1$ pa u tom slučaju imamo

$$\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}q(y|x)f(x) = f(y)q(x|y) = f(y)q(x|y)\rho(y,x).$$

Ako je $\rho(x,y) = 1$, onda je $\rho(y,x) = \frac{f(x)}{f(y)}\frac{q(y|x)}{q(x|y)}$ pa imamo

$$\rho(x,y)q(y|x)f(x) = q(y|x)f(x) = q(y|x)f(x)\frac{f(y)}{f(y)}\frac{q(x|y)}{q(x|y)} = \rho(y,x)q(x|y)f(y).$$

Jasno je da vrijedi:

$$(1 - r(x))\delta_x(y)f(x) = (1 - r(y))\delta_y(x)f(x),$$

jer je $\delta_x(y) = 0$ ako je $x \neq y$.

Dakle, vrijedi uvjet detaljne ravnoteže čime smo dokazali *(i)*. Dio *(ii)* slijedi iz teorema 1.11. □

To da je f stacionarna distribucija povlači da ukoliko je $X_n \sim f$, tada je $X_{n+1} \sim f$, no hoće li ikad doći do toga da u nekom koraku bude $X_n \sim f$? Idući korak je pokazati da Markovljev lanac generiran Metropolis-Hastings algoritmom 1 uistinu konvergira prema stacionarnoj distribuciji f. Prvi korak prema tome je osigurati da je lanac f-ireducibilan. Trebamo osigurati da

$$\forall x \in \mathcal{X}, \quad \forall A \in \mathcal{B}(\mathcal{X}), \quad \int_A f(x)\,\lambda^{(m)}(dx) > 0 \implies \exists n \in \mathbb{N}, \quad K^n(x,A) > 0.$$

Drugim riječima, za svaki skup mjere veće od 0, u odnosu na mjeru definiranu preko gustoće f, vjerojatnost dolaska u taj skup u konačno mnogo koraka mora biti veća od 0. Ukoliko q izaberemo tako da vrijedi

$$q(y|x) > 0, \quad \text{za sve } (x,y) \in \operatorname{supp} f \times \operatorname{supp} f,$$

tada ćemo moći u jednom koraku doći u svaki skup mjere veće od 0 u odnosu na mjeru definiranu preko gustoće f, tj. naš lanac će biti f-ireducibilan. Uz ovu spoznaju sljedeća lema nam govori da je naš lanac i Harris povratan. Nju navodimo bez dokaza. Za dokaz pogledati Robert i Castella [**?**] (lema 7.3).

Lema 2.17. *Ako je lanac dobiven Metropolis-Hastings algoritmom 1 f-ireducibilan, onda je i Harris povratan.*

Također, zbog toga što lanac dobiven Metropolis-Hastings algoritmom omogućuje ostanak u istom stanju, on je aperiodičan. Sada imamo sve uvjete zadovoljene da primijenimo teoreme iz prošlog poglavlja. Dobivene rezultate spojit ćemo u jedan teorem.

Teorem 2.18. *Ukoliko je lanac (X_n) dobiven Metropolis-Hastings algoritmom 1 f-ireducibilan, tada vrijedi:*

(i) f je stacionarna distribucija lanca (X_n),

(ii) za svaki $A \in \mathcal{B}(\mathcal{X})$ i svaku početnu distribuciju μ

$$\mathbb{P}_\mu(X_n \in A) \longrightarrow \int_A f(x)\lambda^{(m)}(dx), \quad n \to \infty,$$

(iii) ako je $h \in L^1(\pi)$, pri čemu je π mjera generirana sa f, tada

$$\hat{I}_N = \frac{1}{N}\sum_{n=1}^{N} h(X_n) \longrightarrow \int_{\mathcal{X}} h(x)f(x)\lambda^{(m)}(dy), \quad n \to \infty,$$

tj. (3) je aproksimacija integrala (2).

Dokaz. Teorem 2.16 povlači tvrdnju (i). Zbog leme 2.17, aperiodičnosti i tvrdnje (i) možemo primijeniti teorem 1.12. Iz njega i raspisa navedenog iza iskaza tog teorema slijedi tvrdnja (ii). Mjera generirana funkcijom f je konačna pa je i σ-konačna zbog čega iz Ergodskog teorema 1.13 i raspisa iza njega slijedi tvrdnja (iii). □

2.4 Primjeri

U nastavku ovog poglavlja pokazat ćemo dva primjera Metropolis-Hastings algoritma: Nezavisni Metropolis-Hastings i Metropolis-Hastings sa slučajnom šetnjom. Komentirat ćemo kako odabir *proposal* distribucije utječe na dobivene procjene integrala (2) na konkretnom primjeru na kojem znamo njegovu vrijednost. No, kako bismo mogli uspoređivati dobivene procjene i kada ne znamo pravu vrijednost integrala, prvo ćemo uvesti dvije numeričke mjere koje će nam pomoći da usporedimo dobivene rezultate.

Efektivna veličina uzorka - ESS

Neka je $(X_n,\, n \geq 0)$ Markovljev lanac dobiven Metropolis-Hastings algoritmom. Slučajne varijable (X_n) nisu nezavisne, što znači da X_n i X_{n+1} daju djelomično redundantnu informaciju prilikom računanja integrala (2). Drugim riječima, lanac duljine 10 000 ne daje toliko mnogo informacija o stacionarnoj distribuciji f kao što to daje

nezavisan uzorak iz te distribucije duljine 10 000. Željeli bismo imati neku mjeru koja nam govori koliko je nezavisne informacije prisutno u lancu. Možemo se pitati kolika je veličina sasvim nekoreliranog lanca koji bi nam dao istu tu informaciju. Odgovor na to pitanje daje nam **efektivna veličina uzorka** (eng. *effective sample size*, ESS). Definira se izrazom

$$ESS = N \Big/ \left(1 + 2\sum_{k=1}^{\infty} \mathrm{acf}(k)\right),$$

pri čemu je N veličina lanca, a $\mathrm{acf}(k)$ je autokorelacija lanca s pomakom k, tj. korelacija između lanca i lanca pomaknutog za k koraka. Od dva lanca, bolji je onaj s većim ESS-om. Efektivnu veličinu lanca uveli su Neal, Carlin i Gelman u radu [2], a dobro objašnjenje može se naći i u Kruschke [3].

Monte Carlo standardna pogreška - MCSE

Monte Carlo standardna pogreška (eng. *Monte Carlo standard error*, MCSE) je standardna devijacija Monte Carlo procjenitelja $\hat{I}_N$ (3). Što je standardna devijacija manja, možemo biti sigurniji da imamo dobru procjenu. Monte Carlo standardnu pogrešku dobivamo preko centralnog graničnog teorema za Markovljeve lance. Centralni granični teorem kaže da

$$\sqrt{N}\left(\underbrace{\frac{1}{N}\sum_{n=0}^{N} h(X_n)}_{\hat{I}_N} - \underbrace{\mathbb{E}_f[h(X_0)]}_{I}\right) \xrightarrow{D} N(0, \sigma_h^2),$$

kada $N \longrightarrow \infty$, pri čemu je

$$\sigma_h^2 = \mathrm{Var}_f[h(X_0)] + 2\sum_{i=1}^{\infty} \mathrm{Cov}[h(X_0), h(X_i)].$$

Indeks f znači da se očekivanja računaju uz pretpostavku $X_0 \sim f$. Za primjenu centralnog graničnog teorema nužno moramo imati ergodski Markovljev lanac sa stacionarnom distribucijom f, što zahtjeva i Metropolis-Hastings algoritam. Postoji mnogo dovoljnih uvjeta, a mi navodimo sljedeća dva (treba vrijediti barem jedan):

- lanac je geometrijski ergodski i $\mathbb{E}_f[|h(X)|^{2+\delta}] < \infty$ za neki $\delta > 0$,
- lanac je uniformno ergodski i $\mathbb{E}_f[h(X)^2] < \infty$.

Ukoliko su zadovoljeni uvjeti centralnog graničnog teorema, za procjenu Monte Carlo standardne pogreške trebamo procijeniti σ_h^2. Postoji više načina kako se to može napraviti, no u to nećemo ulaziti. Za detalje pogledati [1].

2.4.1 Nezavisni Metropolis-Hastings algoritam

Kod nezavisnog Metropolis-Hastings algoritma (eng. *the Independent Metropolis-Hastings algorithm*) *proposal* distribucija q ne ovisi o trenutnom stanju u kojem se lanac (X_n) nalazi, tj. $q(y|x) = q(y)$. Dakle, cijelo vrijeme generiramo uzorak iz iste vjerojatnosne distribucije dane gustoćom $q(y)$. Nužan uvjet kako bi vrijedili rezultati iz prethodnog potpoglavlja je da

$$\operatorname{supp} f \subset \operatorname{supp} q.$$

Algoritam glasi ovako:

Algoritam 2 Nezavisni Metropolis-Hastings algoritam

1: Uz dano $X_n = x_n$
2: Generiraj $Y_n \sim q$
3: Generiraj $U \sim U([0,1])$
4: Izračunaj $\rho(x_n, Y_n)$, pri čemu je

$$\rho(x,y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x)}{q(y)}, 1\right\}$$

5: Definiraj

$$X_{n+1} = \begin{cases} Y_n & \text{ako } U \leq \rho(x_n, Y_n) \\ x_n & \text{inače} \end{cases}$$

6: Ponavljaj za $n = n + 1$

Iako su u algoritmu Y_n generirani nezavisno (od tu i dolazi ime algoritma), dobiveni uzorak nije nezavisan jer vjerojatnost prihvaćanja ρ ovisi o prijašnjem stanju. U nastavku navodimo dovoljan uvjet kako bi dobiveni Markovljev lanac bio uniformno ergodski.

Teorem 2.19. *Algoritam 2 generira uniformno ergodski Markovljev lanac ukoliko postoji konstanta M takva da je*

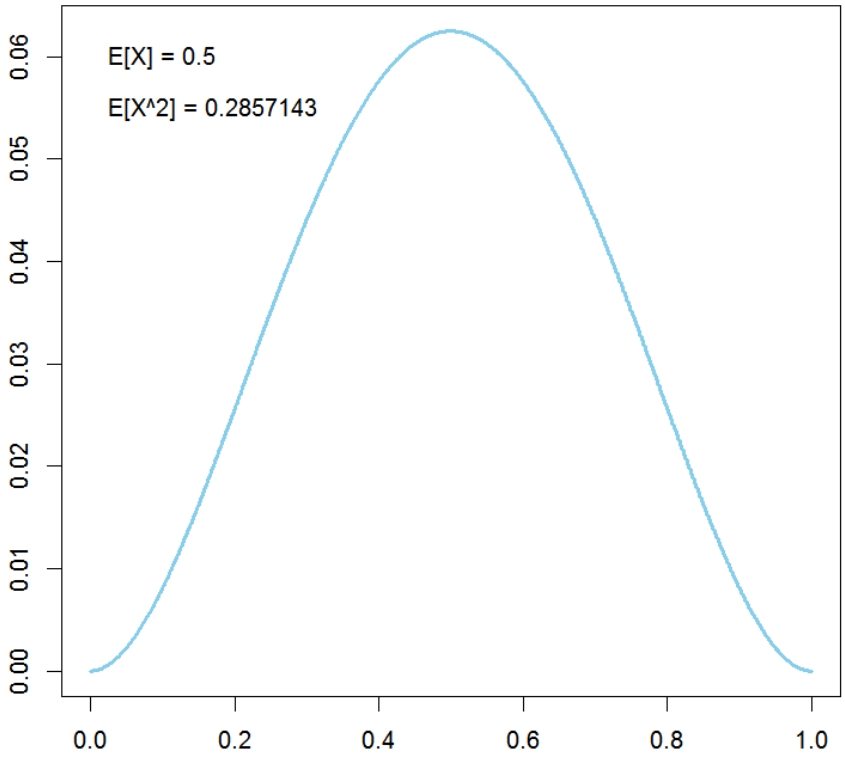
$$f(x) \leq Mq(x), \text{ za svaki } x \in \operatorname{supp} f.$$

U tom slučaju je

$$\|K^n(x,\cdot) - \nu_f(\cdot)\|_{TV} \leq 2\left(1 - \frac{1}{M}\right)^n,$$

pri čemu je ν_f mjera generirana vjerojatnosnom funkcijom gustoće f.

Iako u praksi, ukoliko f znamo samo do na konstantu, nećemo moći izračunati M i dobiti točnu procjenu greške, ovaj teorem nam govori kakvu *proposal* distribuciju q izabrati ukoliko koristimo nezavisni Metropolis-Hastings. Želimo da konstanta M bude što manja, bliža broju 1, pa je najbolje uzeti gustoću q što sličniju gustoći f. Samim time želimo što veći broj prihvaćenih koraka (eng. *acceptance rate*), tj. broj koraka kada Metropolis-Hastings algoritam prihvati vrijednost iz *proposal* distribucije. U idealnom bi slučaju gustoća q bila jednaka gustoći f i svaki korak bismo prihvatili.



Slika 1: Nenormalizirana beta$(3, 3)$ gustoća

Primjer 2.20. *beta*$(3, 3)$
Sada ćemo primijeniti Nezavisni Metropolis-Hastings algoritam za simuliranje uzorka iz beta$(3, 3)$ razdiobe. Napomenimo najprije da ovaj primjer nije realan primjer gdje se koristi Metropolis-Hastings algoritam, no poslužit će kao vrlo jednostavan primjer za ilustraciju teorijskih rezultata dobivenih gore. Gustoća beta(α, β) distribucije dana je izrazom

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \mathbb{1}_{[0,1]},$$

pri čemu je $B(\alpha, \beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\, du$ normalizacijska konstanta. Ovdje već možemo naslutiti kako će za neke višedimenzionalne razdiobe ovu normalizacijsku konstantu biti teško izračunati. Kod Metropolis-Hastings algoritma takve integrale ne moramo znati izračunati te upravo zbog toga on dobiva na važnosti. Ako je $X \sim$ beta(α, β), tada vrijedi

$$\mathbb{E}X = \frac{\alpha}{\alpha+\beta} \quad \text{i} \quad \mathbb{E}[X^2] = \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)}.$$

proposal	U([0,1])	N(0.5,0.24^2)	N(0,1)
ESS	586.3	777.5	137.6
acceptance rate	62.26%	89.99%	21.92%
očekivanje	0.49177	0.50416	0.5217
MCSE	0.00869	0.00506	0.01493
greška	0.00823	0.00416	0.021696
drugi moment	0.27769	0.29209	030229
MCSE	0.00871	0.00546	0.01555
greška	0.00802	0.00637	0.01658

Tablica 1: Rezultati simulacija za različite proposal distribucije

Beta razdioba često se pojavljuje kao osnovni primjer u bayesovskoj statistici i njome se često modelira vjerojatnost uspjeha, npr. dobivanja glave kod bacanja novčića. Mi ćemo simulirati uzorak iz beta$(3, 3)$ razdiobe čija je nenormalizirana gustoća dana na slici 1. Njezino očekivanje je 0.5 dok je drugi moment 0.2857143.

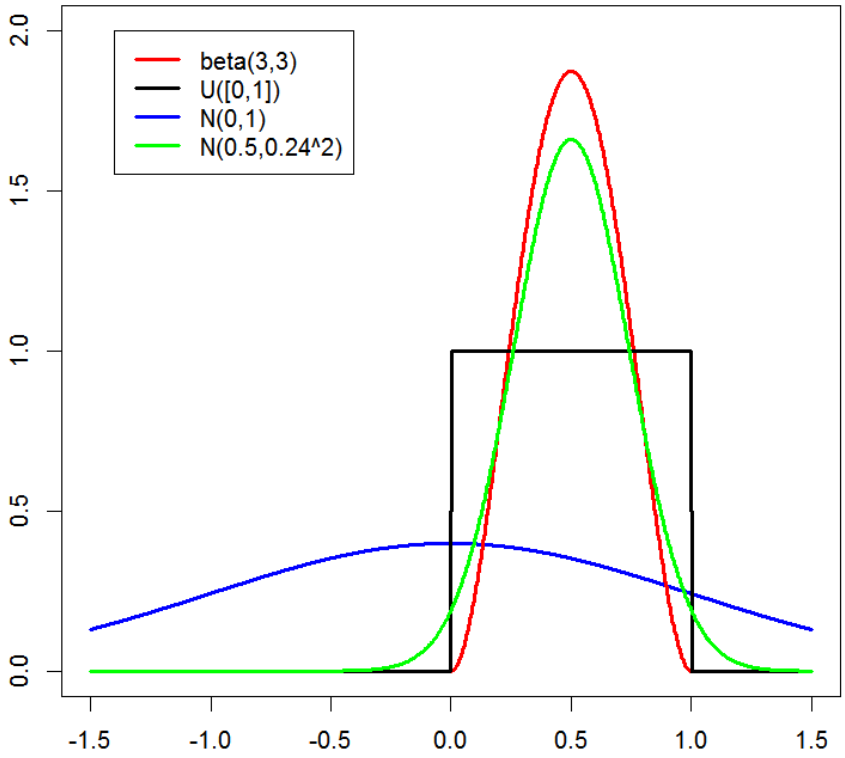
Za *proposal* distribuciju možemo uzeti bilo koju razdiobu kojoj je nosač nadskup od segmenta $[0, 1]$. Usporedit ćemo rezultate kada koristimo tri različite *proposal* distribucije:

- uniformna razdioba na $[0, 1]$ - $U([0, 1])$,
- normalna razdioba s očekivanjem 0.5 i varijancom 0.24^2 - $N(0.5, 0.24^2)$,
- jedinična normalna razdioba - $N(0, 1)$.

Simulacije ćemo napraviti u programskom jeziku R, pri čemu ćemo se praviti da ne znamo normalizacijsku konstantu $B(3, 3)$. Lanci će biti duljine 1 000, a za svaku *proposal* distribuciju izračunat ćemo efektivnu veličinu uzorka, *acceptance rate*, Monte Carlo procjenu očekivanja i drugog momenta te Monte Carlo standardnu pogrešku za te procjenitelje. Također, pošto u ovom umjetnom primjeru znamo prave vrijednosti očekivanja i varijance, usporedit ćemo dobivene procjene s pravim vrijednostima. Rezultati se mogu vidjeti u tablici 1.

Iz tablice vidimo da najtočniju procjenu dobivamo kada za *proposal* distribuciju uzmemo $N(0.5, 0.24^2)$. Efektivna veličina uzorka je najveća, a procjene standardne greške su najmanje. Uniformna razdioba također daje dosta dobre procjene i male greške, dok su kod jedinične normalne razdiobe greške nešto veće. Razlog tomu možemo vidjeti na slici 2.

Vidimo da najbolje rezultate daje ona *proposal* gustoća koja je najsličnija beta$(3, 3)$ gustoći. Što su te gustoće sličnije, konstanta M iz teorema 2.19 je manja kao što možemo vidjeti u tablici 2. Tada imamo bržu konvergenciju pa su i rezultati točniji. U primjeni nije realno da ćemo znati naći tako dobro odgovarajuću *proposal* gustoću kao ovdje, no to nas ne treba previše zabrinjavati. Glavo je imati na umu da ukoliko možemo izabrati *proposal* gustoću sličnu ciljnoj gustoći, to i napravimo. Ukoliko to ne možemo, preciznije rezultate dobit ćemo povećavanjem duljine lanca.



Slika 2: Graf ciljne gustoće i *proposal* gustoća

Duljina lanca u ovom primjeru bila je 1 000 što je zapravo vrlo malo. Ukoliko uzmemo duljinu lanca 10 000, dobit ćemo puno bolje procjene čak i s lošim izborom *proposal* gustoće.

2.4.2 Metropolis-Hastings sa slučajnom šetnjom

Vjerojatno prirodniji odabir prilikom konstrukcije Metropolis-Hastings algoritma je prilikom predlaganja novog stanja uzeti u obzir trenutno stanje. Na taj način lokalno pretražujemo skup stanja. Dakle, novo stanje Y_n biramo prema

$$Y_n = X_n + \epsilon_n,$$

pri čemu je ϵ_n slučajna varijabla s distribucijom q koja ne ovisi o X_n. Na primjer, možemo uzeti $\epsilon_n \sim N(0, \sigma^2)$ i tada ukoliko se u trenutku n nalazimo u stanju x_n, nova predložena vrijednost bit će distribuirana prema $N(x_n, \sigma^2)$. Tada će *proposal* gustoća biti oblika $q(y|x) = q(y - x)$, a Markovljev lanac pridružen toj prijelaznoj jezgri nazivamo slučajna šetnja. Takvu modifikaciju Metropolis-Hastings algo-

	M
U([0,1])	1.87
N(0.5,0.24^2)	1.14
N(0,1)	5.37

Tablica 2: Dobivene konstante M za različite *proposal* distribucije

rimta zovemo **Metropolis-Hastings sa slučajnom šetnjom** (eng. *Random walk Metropolis-Hastings*). Za distribuciju od ϵ_n najčešće se koriste simetrične razdiobe poput uniformne, normalne ili studentove t-distribucije s očekivanjima 0. Ukoliko baš izaberemo simetričnu distribuciju, tada dobivamo orginalan algoritam koji je predložio Metropolis (1953).

Algoritam 3 Metropolis-Hastings algoritam sa slučajnom šetnjom

1: Uz dano $X_n = x_n$
2: Generiraj $Y_n \sim q(|y - x_n|)$
3: Generiraj $U \sim U([0,1])$
4: Izračunaj $\rho(x_n, Y_n)$ pri čemu je

$$\rho(x,y) = \min\left\{\frac{f(y)}{f(x)}, 1\right\}$$

5: Definiraj

$$X_{n+1} = \begin{cases} Y_n & \text{ako } U \leq \rho(x_n, Y_n) \\ x_n & \text{inače} \end{cases}$$

6: Ponavljaj za $n = n + 1$

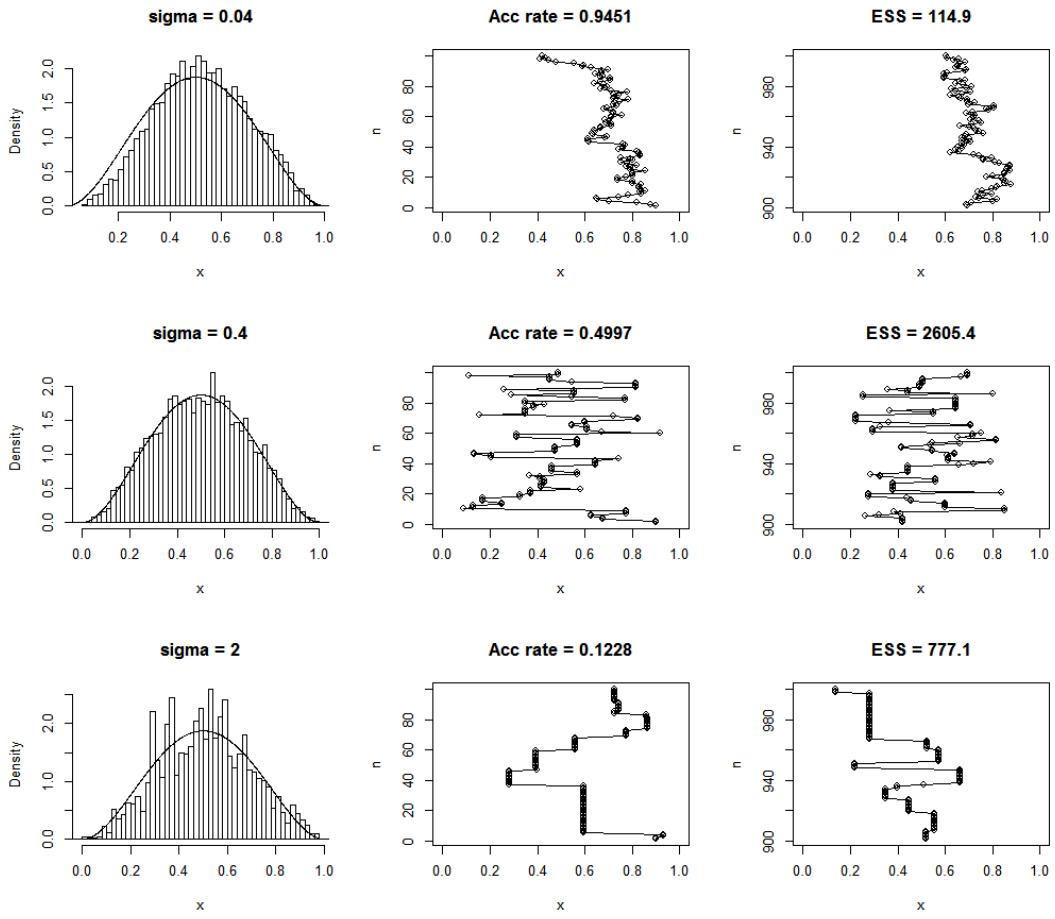
Vidimo da se izraz ρ pojednostavio. Sada u predloženo stanje sigurno prelazimo ukoliko je gustoća u tom stanju veća, a ukoliko je manja, prelazimo s vjerojatnosti koja ovisi o tome koliko je gustoća u predloženom stanju manja.

Nažalost, Metropolis-Hastings algoritam sa slučajnom šenjom ne mora generirati lanac koji je uniformno ergodski. Mengersen i Tweedie pokazali su da ukoliko je nosač ciljne gustoće skup svih realnih brojeva, tada ni ne može generirati lanac koji je uniformno ergodski. Ipak, mogu se izvesti nužni i dovoljni uvjeti kako bi generirani lanac bio geometrijski ergodski. Za detalje pogledati Robert, Casella [6] (poglavlje 7.5).

Za precizne rezultate kod primjene ovog algoritma često je potrebno skalirati *proposal* gustoću, što ćemo ilustrirati sljedećim primjerom.

Primjer 2.21. beta(3, 3)
Ciljna distribucija bit će nam beta(3, 3) kao i u prethodnom primjeru, no ovoga puta koristit ćemo Metropolis-Hastings algoritam sa slučajnom šetnjom. Ukoliko se u trenutku n nalazimo u stanju x_n, tada će nam Y_n biti distribuiran prema $N(x_n, \sigma^2)$, pri čemu ćemo promatrati različit izbor standardne devijacije σ. Mijenjanje standardne

devijacije odgovara skaliranju skokova. Promatrat ćemo vrijednosti $\sigma = 0.04, 0.4, 2$. Za svaku od tih vrijednosti nacrtat ćemo histogram relativnih frekvencija dobivenog uzorka i usporediti ga s gustoćom beta$(3, 3)$ razdiobe. Također, nacrtat ćemo prvih 100 koraka dobivenog lanca kao i posljednjih 100. Lanci će biti duljine 10 000 koraka. Lance pokrećemo iz početnog stanja $x_0 = 0.95$. Rezultate možemo vidjeti na slici 3.



Slika 3: Histogram relativnih frekvencija, početak i kraj lanaca za različit izbor standardne devijacije

Ako je standardna devijacija premala (slučaj $\sigma = 0.04$), tada *proposal* distribucija predlaže jako male pomake. Tada je $\frac{f(Y_n)}{f(x_n)}$ blizu 1 pa jako često prihvaćamo predložene korake. No, kako su koraci jako blizu, treba nam puno vremena da se maknemo iz dijela prostora stanja gdje ciljna gustoća nema veliku masu. To možemo vidjeti na slici 3 u prvom retku i drugom stupcu. Trebalo nam je 100 koraka da dođemo do dijela gdje ciljna gustoća ima veliku masu. Samim time pretraživanje prostora stanja vrlo je sporo, a informacije koje dobivamo su redundantne. Zato je efektivna veličina uzorka vrlo mala. Dakle, zaključujemo da kod slučajne šetnje visok *acceptance rate* nije dobar, kao što je to bio kod Nezavisnog Metropolis-Hastings algoritma.

Ako je standardna devijacija prevelika (slučaj $\sigma = 2$), tada *proposal* distribucija često predlaže vrlo velike skokove koji izlaze iz nosača ciljne gustoće pa su automatski odbijeni. To znači da lanac ostaje u istom stanju dugo vremena te samim time donosi redundantne informacije. Zato histogram ima puno stupaca koji su iznad ciljne gustoće. U ovom slučaju *acceptance rate* vrlo je malen.

U slučaju kada je $\sigma = 0.4$, vidimo da se histogram dosta dobro slaže s ciljnom gustoćom. Lanac uspijeva dosta dobro pretražiti prostor stanja, a *acceptance rate* je oko 50%. Vidimo da su ESS i *acceptance rate* dobri numerički pokazatelji koliko smo uspješno skalirali našu *proposal* distribuciju.

Spomenuli smo da lancu često treba dosta koraka da dođe iz područja male mase u područje velike mase ciljne gustoće. Često se zbog toga odbacuje početni dio lanca. Taj dio lanca zovemo *burn-in* period. Ideja je da informacije počnemo uzimati tek kad lanac dođe u područje veće mase kako ne bismo imali previše točaka iz područja gdje ciljna gustoća nema veliku masu.

2.5 Gibbsovo uzorkovanje

Metropolis-Hastings algoritam vrlo je općenit i široko primjenjiv. Problem je što *proposal* distribucija mora biti dobro usklađena s ciljnom distribucijom kako bi algoritam bio efikasan. No, postoje i druge metode kao što je **Gibbsovo uzorkovanje** (eng. *Gibbs sampler*). Gibbsovo uzorkovanje uveli su Geman i Geman 1984. godine, a metoda je dobila ime po fizičaru Josiahu Willardu Gibbsu. Algoritam Gibbsovog uzorkovanja zasniva se na slučajnoj šetnji kroz prostor parametara, kao i Metropolis-Hastings algoritam. Pretpostavimo da želimo simulirati uzorak iz višedimenzionalne razdiobe s gustoćom $f(x_1, x_2, \ldots, x_m)$.

Algoritam 4 Gibbsovo uzorkovanje

1: Inicijaliziraj lanac početnim stanjem $(x_1^{(0)}, \ldots, x_m^{(0)})$; pretpostavimo da smo u n-tom koraku došli u stanje $(x_1^{(n)}, \ldots, x_m^{(n)})$
2: Simuliraj $x_1^{(n+1)}$ iz $f(x_1|x_2^{(n)}, \ldots, x_m^{(n)})$, $x_2^{(n+1)}$ iz $f(x_2|x_1^{(n+1)}, x_3^{(n)}, \ldots, x_m^{(n)})$ i tako sve do $x_m^{(n+1)}$ iz $f(x_m|x_1^{(n+1)}, \ldots, x_{m-1}^{(n+1)})$
3: Ponavljaj za $n = n + 1$

Šetnja kreće iz neke proizvoljne točke, a svaki korak ovisi samo o trenutnom stanju u kojem se nalazimo. U svakom koraku izabiremo jedan parametar iz ciljne gustoće. Tipično se ide kružno od x_1 pa sve do x_m i tako dalje. Razlog tomu je što bi kod velikih modela s mnogo parametara previše koraka bilo potrebno da bi se svi parametri izabrali dovoljan broj puta ako bismo ih birali slučajno. Nakon što smo izabrali x_i, tada novu točku biramo tako da sve ostale točke držimo fiksirane, dok i-tu koordinatu simuliramo iz uvjetne gustoće $f(x_i|x_1^{(n+1)}, \ldots, x_{i-1}^{(n+1)}, x_{i+1}^{(n)}, \ldots, x_n^{(n)})$ ciljne distribucije. Zato su koraci uvijek u smjeru koordinatnih osi. Glavna pretpostavka Gibbsovog uzorkovanja je da znamo simulirati iz uvjetne ciljne distribucije što je u

mnogo statističkih problema moguće. Detaljnije o Gibbsovom uzorkovanju može se naći u Casella [6].

Poglavlje 3

Primjene u bayesovskoj statistici

3.1 Osnovne bayesovske statistike

U klasičnoj statistici dane podatke pokušavamo opisati statističkim modelom koji se sastoji od vjerojatnosnih funkcija distribucija s pripadnim parametrima. Te parametre smatramo unaprijed određenima i fiksiranima, a iz podataka ih pokušavamo procijeniti. Često se parametri procjenjuju metodom maksimalne vjerodostojnosti (eng. *maximum likelihood estimator, MLE*). Na taj način nikakva prijašnja znanja o parametrima od interesa nisu ugrađena u naš model. Kod **bayesovske statistike** (eng. *Bayesian statistics*) osnovna je ideja gledati parametre kao slučajne varijable. Dakle, oni nisu fiksirani već imaju svoju vjerojatnosnu razdiobu. Točnije, uz njih vežemo dvije bitne razdiobe. Prva je **apriori** distribucija (eng. *prior distribution*) koja predstavlja naše znanje o parametru prije nego što smo vidjeli podatke, a druga je **aposteriori** distribucija (eng. *posterior distribution*) koja predstavlja naše znanje o parametru nakon što smo vidjeli podatke. Apriori i aposteriori distribucija povezane su preko **Bayesovog pravila**:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto p(D|\theta)p(\theta), \tag{4}$$

pri čemu D predstavlja podatke, θ parametar koji promatramo, $p(\theta)$ apriori distribuciju parametra, $p(\theta|D)$ aposteriori distribuciju parametra, a $p(D|\theta)$ vjerodostojnost (eng. *likelihood*) podataka uz dani model. Oznaka $\propto$ znači da su izrazi proporcionalni, tj. da vrijedi

$$p(\theta|D) = c\, p(D|\theta)p(\theta),$$

za neki $c \in \mathbb{R}$ i vrlo je česta u literaturi o bayesovskoj statistici. Normalizacijska konstanta $p(D)$ može se izraziti kao

$$p(D) = \int p(D|\theta)p(\theta)\, d\theta,$$

no u praksi ju može biti vrlo teško izračunati. Iz (4) vidimo da u zaključivanju o parametru kojeg promatramo sudjeluju i prijašnje znanje (kroz apriori distribuciju) i podaci koje smo prikupili (kroz vjerodostojnost). Iako je korisno u model uključiti prijašnje znanje o parametru, bayesovska statistika je često i kritizirana zbog toga jer razne osobe mogu koristiti različite apriori distribucije i time dobiti različite rezultate i zaključke. Time dolazi do problema konzistentnosti ovog pristupa. No, postoje mnoge metode i upute kako izabrati prikladnu apriori distribuciju te načini kako se može promatrati utjecaj izbora apriori distribucije na krajnje rezultate. Prilikom provođenja istraživanja vrlo je bitno uključiti stručnjake iz dane domene te javno raspravljati o izboru apriori distribucije. Tek nakon što se postigne dogovor oko apriori distribucije, ima smisla provoditi bayesovsku statistiku. U nastavku donosimo jednostavan primjer kako bismo ilustrirali uvedene pojmove.

Primjer 3.22. Bacanje novčića

Zamislimo da se šećemo ulicom i nađemo novčić. Zanima nas vjerojatnost dobivanja glave na tom novčiću. Označimo tu vjerojatnost sa θ. Prvo trebamo odrediti apriori distribuciju za θ. Kako je vjerojatnost dobivanja glave broj u segmentu $[0, 1]$, trebamo zadati neku gustoću koja ima nosač na $[0, 1]$. Zamislit ćemo tri moguće situacije. U prvoj prepoznajemo da je novčić zapravo kovanica od 5 kuna. Takvih novčića već smo puno vidjeli u svome životu i smatramo da je vjerojatnost dobivanja glave oko 50% pa ćemo za apriori distribuciju izabrati beta$(5, 5)$ razdiobu. U drugom scenariju zamislimo da smo našli novčić na kojem piše "trik novčić". Za takav novčić bismo mogli znati da preferira jednu stranu (glavu ili pismo), tj. da je nepošten, ali ne znamo koju. U tom slučaju mogli bismo pretpostaviti beta$(0.9, 0.9)$ apriori distribuciju. U trećem slučaju zamislimo da smo našli novčić bez nekih natpisa i ne znamo nikakve informacije o tom novčiću. Tada želimo neku neinformativnu apriori distribuciju (eng. *noninformaive prior*) pa možemo uzeti uniformnu distribuciju na $(0, 1)$. Primijetimo da je $U(0, 1) = \text{beta}(1, 1)$.

Nakon što smo odredili apriori distribucije krećemo s eksperimentom. Pretpostavimo da smo bacili novčić i da smo od $N = 24$ bacanja dobili $z = 8$ glava. Izvedimo aposteriori distribuciju pomoću Bayesovog pravila. Kako su sve tri apriori distribucije beta razdiobe, račun ćemo provesti općenito za beta(α, β) distribuciju:

$$p(\theta|D) = \frac{p(D|\theta)\, p(\theta)}{p(D)} \propto \theta^z (1-\theta)^{N-z}\, \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

iz čega slijedi

$$p(\theta|D) \propto \theta^{\alpha-1+z}(1-\theta)^{\beta-1+N-z}.$$

Kako bismo dobili točnu aposteriori distribuciju, moramo izračunati

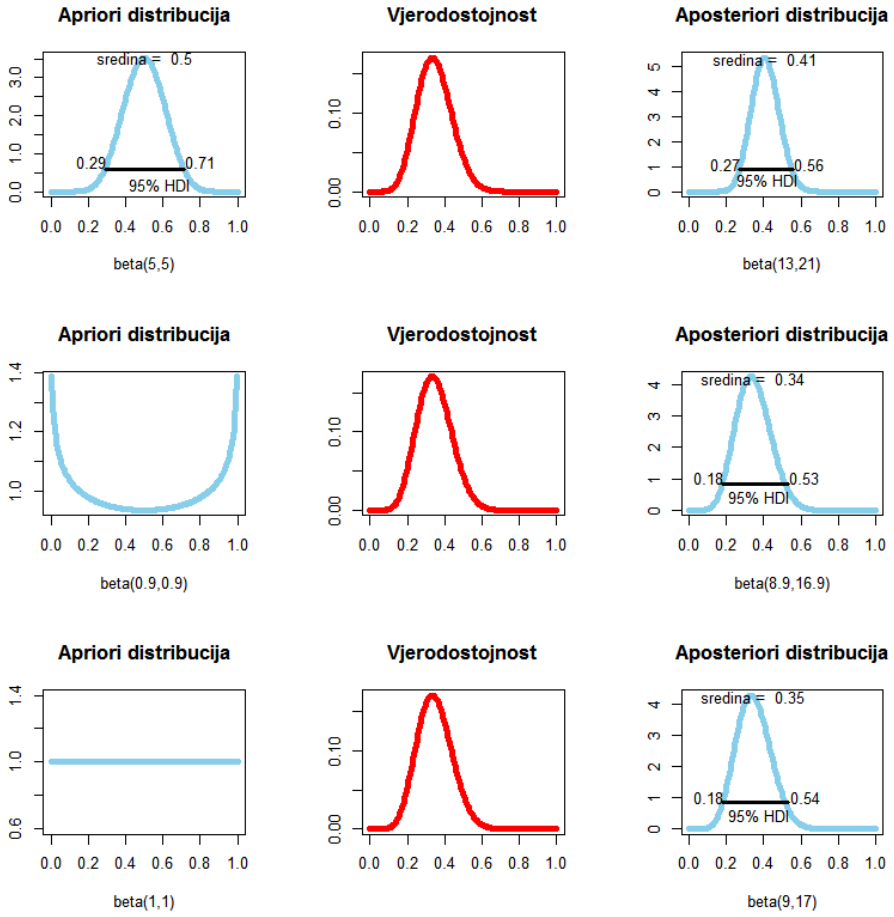
$$p(D) = \int_0^1 \theta^{\alpha-1+z}(1-\theta)^{\beta-1+N-z}\, d\theta.$$

Iako općenito ovaj izraz može biti teško izračunati, u ovom primjeru primjećujemo da je to upravo beta funkcija izračunata u točkama $\alpha+z$ i $\beta+N-z$, tj. $B(\alpha+z, \beta+N-z)$ pa je naša aposteriori distribucija $p(\theta|D)$ jednaka gustoći beta$(\alpha + z, \beta + N - z)$ razdiobe, tj.

$$\theta|D \sim \text{beta}(\alpha + z, \beta + N - z).$$

Kada se dogodi da apriori i aposteriori distribucija pripadaju istoj familiji distribucija, tada kažemo da su one **konjugirane distribucije** za dani model. To svojstvo zapravo ovisi o tome slažu li se vjerodostojnost i apriori distribucija. Često se može naći popis funkcija vjerodostojnosti i odgovarajućih apriori distribucija tako da apriori i aposteriori distribucija budu konjugirane. U našem slučaju bacanje novčića prati Bernoullijevu razdiobu s parametrom θ, tj. uzastopna bacanja su binomno distribuirana s vjerojatnosti uspjeha θ i brojem ponavljanja N. Dakle, binomna funkcija vjerodostojnosti i beta apriori distribucija rezultiraju beta aposteriori distribucijom.

Vratimo se sada na primjer. U eksperimentu smo u $N = 24$ bacanja dobili $z = 8$ glava u sva tri scenarija. Rezultate eksperimenta možemo vidjeti na slici 4. Za aposteriori distribuciju nacrtali smo i 95% interval najveće gustoće (eng. *highest density interval*, HDI).



Slika 4: Rezultati eksperimenta za tri različite apriorne distribucije; plavom bojom označene su vjerojatnosne funkcije gustoće, a crvenom bojom vjerodostojnost

Interval najveće gustoće (HDI) sadrži 95% mase gustoće i za svaku točku u njemu vrijedi da je gustoća veća nego za svaku točku koja nije u njemu. Na taj način dobivamo najmanji pouzdani interval. Za razliku od klasične statistike, interpretacija pouzdanih intervala u bayesovskoj statistici je direktna: vjerojatnost da je parametar u 95% pouzdanom intervalu je 95%. Iz slike vidimo da je u prvom primjeru (kada smo našli kovanicu od 5 kuna) HDI uži nego u ostala dva primjera. To je rezultat toga što vjerujemo da je prava vjerojatnost padanja glave negdje oko 50% pa jako male vrijednosti za θ nisu jako vjerojatne, kao što su u situacijama dva i tri. U njima vidimo da aposteriori distribucija jako sliči na vjerodostojnost, što znači da apriori distribucije nisu imale prevelik utjecaj. Također, očekivanje ili sredina parametra

jako je blizu MLE procjene parametra koja iznosi $\frac{z}{N} = \frac{8}{24} \approx 0.33$. Cilj ovog primjera bio je ilustrirati osnovne pojmove bayesovske statistike pa nećemo ulaziti u to kako testirati vrijednosti parametra θ i donositi binarne odluke. Zainteresiranog čitatelja upućujemo na Kruschke [3].

Općenito, ako nemamo konjugirane distribucije, što je često slučaj, da bismo dobili rezultate eksperimenta, morali bismo izračunati

$$p(D) = \int p(D|\theta)p(\theta)\, d\theta. \tag{5}$$

Integral u (5) može biti jako teško izračunati zbog oblika funkcije vjerodostojnosti i apriori distribucije. Također, model može sadržavati puno parametara pa je integral (5) u visokodimenzionalnom prostoru i teško se može izračunati, čak i aproksimirati numeričkim metodama. Zato se odustaje od ideje računanja $p(D)$ te se traži drugačije rješenje. Jedno rješenje moglo bi biti uzeti gustu mrežu na nosaču aposteriori distribucije, izračunati gustoću u tim točkama i zatim normirati sumom gustoća u tim točkama. Na taj način zapravo diskretiziramo aposteriori distribuciju i aproksimiramo vrijednosti parametara. Ovaj pristup nije dobar jer nosač nekog parametra može biti neograničen. Osim toga, čak i da su nosači svih parametara ograničeni, ukoliko imamo recimo 20 parametara (što i nije puno u nekim primjenama) i uzmemo samo 50 točaka za svaki parametar, dobivamo 10^{33} točaka što je iznimno zahtjevna zadaća za računalo, a aproksimacija bi bila prilično gruba.

Postoji i bolje rješenje problema. Zamislimo da znamo uzorkovati iz aposteriori distribucije. Tada bismo uzimanjem dovoljno velikog uzorka mogli procijeniti očekivanje, varijancu distribucije, pouzdane intervale i sve druge karakteristike distribucije. Drugim riječima, ukoliko imamo jako velik uzorak iz distribucije, to je kao da znamo i samu distribuciju. Na sreću, MCMC algoritmi omogućuju nam upravo to - da simuliramo uzorak iz $p(\theta|D)$, pri čemu je $p(\theta|D)$ dovoljno znati samo do na konstantu, tj. ne trebamo računati $p(D)$. Zbog toga se oni primjenjuju u bayesovskoj statistici. Ideja bayesovske statistike nije nova i moderna (Thomas Bayes živio je u 18. stoljeću), no upravo je razvoj računala i MCMC algoritama učinio bayesovsku statistiku popularnom. Omogućio je njezinu primjenu na kompliciranije, hijerarhijske modele koji mogu dobro opisivati podatke iz stvarnosti. U nastavku rada pokazat ćemo primjer kompliciranijeg modela čije rezultate ne bismo mogli dobiti bez primjene MCMC algoritama.

3.2 Generalizirani linearni modeli

Generalizirani linearni modeli su klasa modela koja ne pripada direktno u bayesovsku statistiku, već je izvorno nastala u frekventističkoj statistici. Razlika je u načinu procjene parametara. U ovom potpoglavlju ukratko ćemo objasniti generalizirane linearne modele i procjenu parametara iz perspektive bayesovske statistike.

Glavni je cilj mnogih istraživanja utvrditi postoji li veza između neke odabrane varijable i ostalih varijabli te ju odrediti. Takva veza rijetko je deterministička pa

ju najčešće modeliramo koristeći vjerojatnosne modele. Odabranu varijablu zovemo **ovisnom varijablom** ili **odzivom** (eng. *response*), a ostale varijable zovemo **neovisnim varijablama, predviditeljima** ili **kovarijatama** (eng. *predictors, features*). Podatke zapisujemo u obliku

$$(\mathbf{x_i}, y_i), \quad i = 1, 2, \ldots, n$$

pri čemu su y_i realizacije ovisne varijable Y_i za čiju razdiobu pretpostavljamo da ovisi o vrijednostima neovisnih varijabli $\mathbf{x_i}$. Neovisnih varijabli može biti proizvoljno mnogo, a mogu biti numeričkog ili kategorijalnog tipa. Često neovisne varijable zapisujemo matrično pri čemu svaki redak predstavlja jednu opservaciju, a stupci neovisne varijable. Neka je n broj opservacija, a m broj neovisnih varijabli. Tada je $X \in \mathbb{R}^{n \times m}$ matrica neovisnih varijabli ili kovarijata. Zapis x_{ij} predstavlja vrijednost j-te neovisne varijable kod i-te opservacije.

Kod linearnih modela osnovna pretpostavka je da postoji linearna veza između očekivanja ovisne varijable i neovisnih varijabli, tj.

$$\mathbb{E}Y_i = \sum_{j=1}^{m} \beta_j x_{ij},$$

pri čemu su β_j parametri modela. Kod generaliziranih linearnih modela pretpostavljena veza ne mora biti linearna, već je neka funkcija od linearne kombinacije zavisnih varijabli. Preciznije, generalizirani linearni model definiran je u tri koraka:

1. Linearni predviditelj: $\sum_{j=1}^{m} \beta_j x_{ij}$.
2. Funkcija veze (eng. *link function*) g koja bijektivno povezuje linearni prediktor s očekivanjem ovisne varijable Y_i,
$$\mathbb{E}Y_i = g^{-1}\left(\sum_{j=1}^{m} \beta_j x_{ij}\right).$$
3. Za zadano očekivanje ovisna varijabla Y_i ima unaprijed određenu razdiobu iz eksponencijalne familije.

Napomenimo da je funkcija g^{-1} inverz funkcije veze g. Tradicionalno funkcija veze g je zamišljena da transformira vrijednosti ovisne varijable u oblik koji se može povezati s linearnim modelom, tj.

$$g\left(\mathbb{E}Y_i\right) = \sum_{j=1}^{m} \beta_j x_{ij}.$$

Od tu dolazi i njezin naziv. Iz ova tri koraka vidimo i otkuda su generalizirani linearni modeli dobili svoje ime. Oni generaliziraju direktnu linearnu vezu između očekivanja ovisne varijable i neovisnih varijabli, ali u sebi i dalje sadrže linearni prediktor.

Primjer 3.23. Linearna regresija

Ukoliko pretpostavimo vezu između ovisne varijable i neovisnih varijabli na sljedeći način

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_m x_{im} + \epsilon_i,$$

pri čemu su $\epsilon_i \sim N(0, \sigma^2)$ i međusobno nezavisne, tada smo dobili model višestruke linearne regresije. Primijetimo da imamo linearnu kombinaciju neovisnih varijabli (korak 1.). Vrijedi

$$\mathbb{E}Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_m x_{im},$$

pa je *link* funkcija identiteta (korak 2.). Iz dane veze slijedi da je

$$Y_i \sim N\left(\sum_{j=0}^{m} \beta_j x_{ij}, \sigma^2\right),$$

uz dogovor $x_{i0} = 1$ pa je i razdioba od Y_i određena (korak 3.).

Primjer 3.24. Logistička regresija

Ukoliko je ovisna varijabla binarna ($Y_i \in \{0, 1\}$), možemo ju modelirati kao Bernoullijevu slučajnu varijablu s očekivanjem $p_i = g^{-1}\left(\sum_{j=1}^{m} \beta_j x_{ij}\right)$. Za inverznu *link* funkciju možemo uzeti sigmoidu $\sigma : \mathbb{R} \to [0, 1]$ definiranu izrazom

$$\sigma(t) = \frac{1}{1 + e^{-t}}, \quad t \in \mathbb{R}.$$

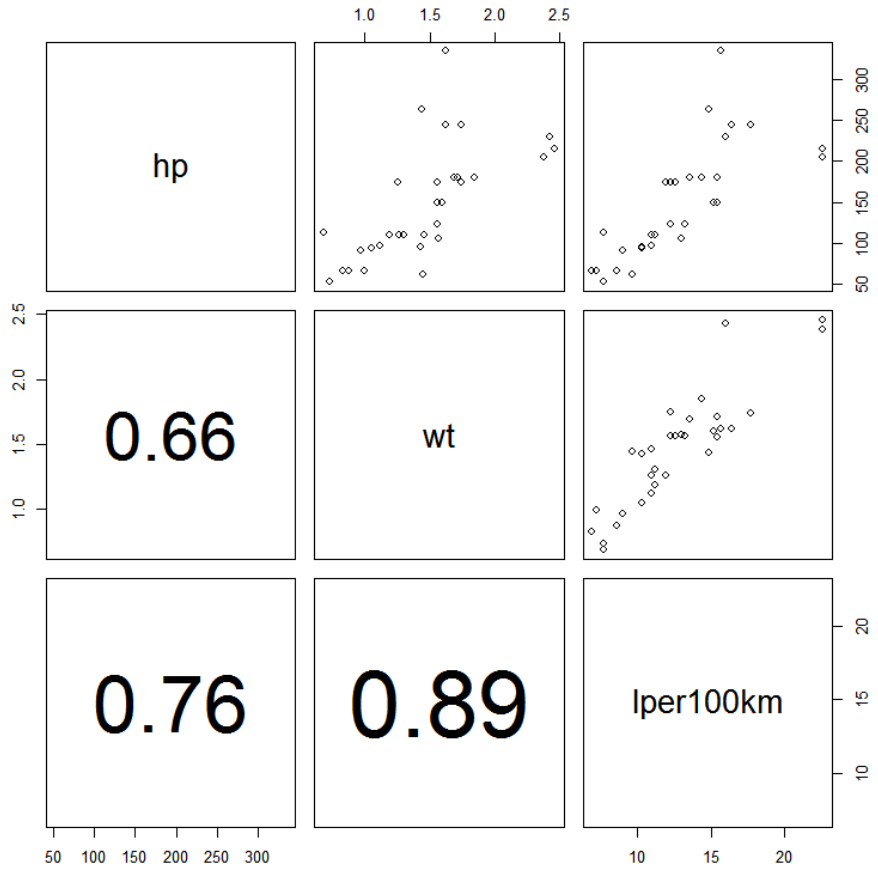
Za procjenu parametara generaliziranih linearnih modela standardno se koristi procedura maksimalne vjerodostojnosti. Podsjetimo se, vjerodostojnost je funkcija koja za dane parametre daje vjerojatnost skupljenih podataka. Kao procjena parametara uzimaju se oni parametri koji maksimiziraju vjerodostojnost. Dakle, procjena parametara vodi na optimizacijski problem. U praksi to može biti vrlo težak optimizacijski problem koji se često rješava numeričkim algoritmima kao što su gradijentna metoda (eng. *gradient descend*) ili Newton-Rapshon metoda. Također, treba voditi i računa da ne dođe do prenaučenosti (eng. *overfitting*) modela pri čemu model jako dobro opisuje dane podatke, ali jako loše radi na neviđenim podacima.

3.3 Primjer: Linearna regresija s teškim repovima

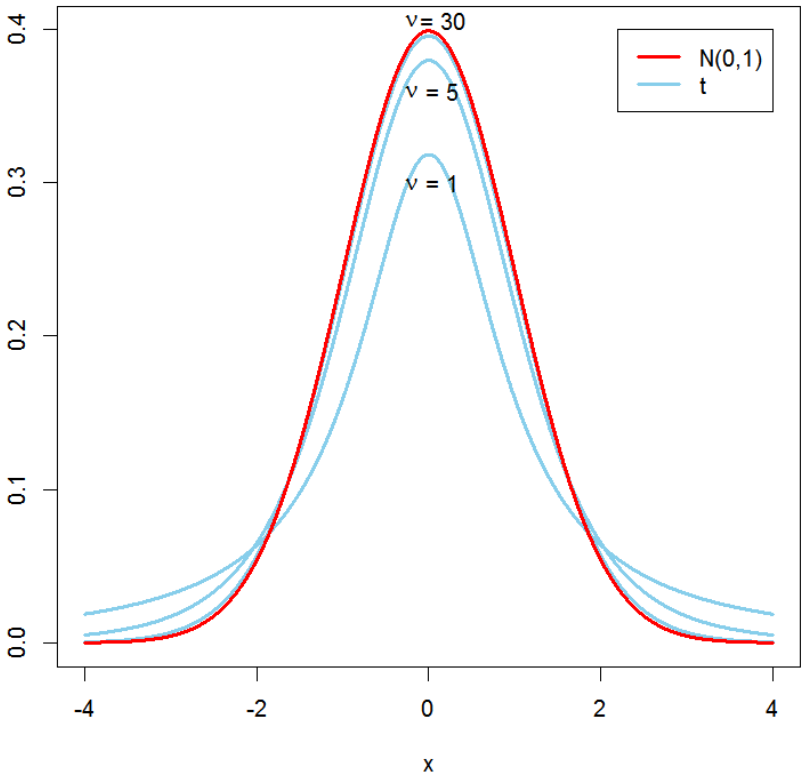
Sada ćemo primijeniti sve o čemu smo do sada pričali na primjeru određivanja potrošnje goriva automobila prema težini automobila i broju konjskih snaga. Sagradit ćemo model linearne regresije s teškim repovima na bayesovski način te ćemo komentirati bitne stvari na koje se treba paziti prilikom gradnje ovakvog modela. Glavna je poanta primjera ilustrirati metode o kojima smo govorili u prethodnim poglavljima, a ne na samim dobivenim rezultatima. Cijeli postupak provest ćemo pomoću programskih jezika R i JAGS. Kratak uvod u JAGS i o tome kako smo dobili rezultate može se naći u *Dodatku*.

Podaci

Podaci koje ćemo koristiti izvučeni su iz automobilskog magazina *Motor Trend US* iz 1974. godine. Sastoje se od 32 modela automobila iz 1973. i 1974. godine, a sadrže karakteristike poput prijeđenih milja po galonu, broja cilindara, konjskih snaga, težine i slično. Mi ćemo koristiti samo broj prijeđenih milja po galonu, broj konjskih snaga i težine automobila radi jednostavnosti. Kako su američke mjerne jedinice različite od europskih, pretvorili smo ih u nama poznatije mjerne jedinice kako bismo imali bolji osjećaj za dobivene brojeve. Točnije, težinu automobila preračunali smo u tone, dok smo broj prijeđenih milja po galonu pretvorili u broj litara potrošenih na 100 kilometara prijeđenih. Prikaz podataka možemo vidjeti na slici 5.



Slika 5: Prikaz podataka u koordinatnom sustavu i korelacije, hp - broj konjskih snaga, wt - težina, lper100km - potrošnja goriva na 100 km



Slika 6: Graf gustoće t-distribucije za različite parametre normalnosti ν i usporedba s jediničnom normalnom distribucijom

Generalizirana studentova t-distribucija

Jedna od distribucija koju ćemo koristiti u našem modelu je i generalizirana t-distribucija. Standardna t-distribucija zadana je brojem stupnjeva slobode, parametrom koji ćemo označavati sa ν. Da bi postojalo očekivanje t-distribucije, ν mora biti veći ili jednak 1. U tom slučaju očekivanje je 0. Što je parametar ν manji, repovi t-distribucije su teži, tj. veća je vjerojatnost većeg odstupanja od očekivanja. Zato se t-distribucija često koristi kod simuliranja rijetkih događaja. Što je parametar ν veći, to t-distribucija više sliči normalnoj. Zato parametar ν možemo zvati i parametar normalnosti. To možemo vidjeti na slici 6. Osim toga, t-distribucija koristi se i kod podataka s puno tzv. *outliera* jer je zbog teških repova procjena očekivanja manje osjetljiva u odnosu na normalnu razdiobu. Zbog toga što ćemo odstupanje od očekivanja modelirati generaliziranom t-distribucijom, a ne normalnom distribucijom kao u običnoj linearnoj regresiji, ovakav model zovemo linearna regresija s teškim repovima. Generalizirana t-distribucija ima još dva parametra.

Neka je $X \sim t(\nu)$. Tada se generalizirana studentova t-distribucija definira kao

razdioba slučajne varijable

$$T = \mu + \sigma X, \quad \text{u oznaci} \quad T \sim t(\nu; \mu, \sigma).$$

Parametar μ zove se lokacijski parametar (eng. *location parameter*), a u našem slučaju on je ujedno i očekivanje. Parametar σ zove se parametar skaliranja (eng. *scale parameter*) i odmah naglašavamo da σ nije standardna devijacija generalizirane t-distribucije. Detaljnije razmatranje značenja parametara ν, μ i σ može se naći u Krushke [3], poglavlje 16.2.

Definiranje modela i apriori distribucija

U ovom primjeru modeliramo ovisnost ovisne varijable Y_i koja predstavlja potrošnju automobila na prijeđenih 100 kilometara. Zavisne varijable su broj konjskih snaga i težina automobila. Konstruiramo generalizirani linearni model tako da očekivanje ovisne varijable procjenjujemo direktno linearnom kombinacijom nezavisnih varijabli (*link* funkcija g je identiteta), dok za zadano očekivanje Y_i ima generaliziranu studentovu t-distribuciju s parametrima ν i σ. Dakle,

$$Y_i \sim t(\nu; \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}, \sigma).$$

U bayesovskoj statistici parametre distribucija smatramo slučajnim varijablama pa je idući korak odrediti apriori distribucije na parametre ν, σ, β_0, β_1 i β_2.

Za parametar normalnosti ili broj stupnjeva slobode ν rekli smo da utječe na repove t-distribucije. Što je veći, to t-distribucija više sliči normalnoj razdiobi. Pravilo desne ruke kaže da je za $\nu \geq 30$ t-distribucija poprilično slična normalnoj razdiobi (velike varijacije u izgledu t-distribucije događaju se za male vrijednosti parametra ν) kao što možemo vidjeti na slici 6. Zato želimo uzeti apriori distribuciju koja daje mogućnost parametru ν da poprima male vrijednosti kao i velike. Kako mora vrijediti $\nu \geq 1$, prvo ćemo napraviti transformaciju $\nu' = \nu - 1$. Zatim ćemo za apriori razdiobu od ν' koristiti eksponencijalnu razdiobu s parametrom $\frac{1}{29}$, tj.

$$\nu' \sim \text{Exp}\left(\frac{1}{29}\right).$$

Očekivanje te razdiobe je 29 pa je očekivana vrijednost za ν upravo 30. Iako je puno mase stavljeno na vrijednosti manje od 30 (za ν), zastupljene su i velike vrijednosti jer je očekivanje 30. Ukoliko u aposteriori distribuciji ν poprima male vrijednosti, onda u podacima možemo očekivati dosta *outliera* i koristimo svojstvo t-distribucije da ima teške repove, dok ukoliko ν bude veći od 30, onda smo blizu konteksta obične linearne regresije s normalnim šumom. Točna vrijednost parametra ν nije nam previše bitna. Važnije nam je da smo dali modelu fleksibilnost da koristi teške repove ukoliko podaci tako nalažu.

Za parametar skaliranja σ uzet ćemo uniformnu distribuciju na širokom segmentu:

$$\sigma \sim U([10^{-5}, 10^5]).$$

To je primjer tzv. neinformativne apriori distribucija i prilikom množenja funkcijom vjerodostojnosti, aposteriori distribucija bit će vrlo slična vjerodostojnosti. Kako bismo potvrdili da je to uistinu neinformativna apriori distribucija, za apriori distribuciju od σ koristit ćemo i gamma distribuciju s očekivanjem 1 i standardnom devijacijom 1 000 te ćemo usporediti aposteriori distribuciju za σ u ovisnosti te dvije apriori distribucije.

Prije nego što odredimo apriori distribuciju za parametre $\beta_0, \beta_1, \beta_2$, zamislimo da pravcem pokušavamo opisati dvodimenzionalne podatke koji se nalaze daleko od središta koordinatnog sustava (možemo zamišljati da se nalaze recimo u kvadratu $[50, 51] \times [50, 51]$). Tada imamo parametre odsječak na osi y (β_0) i nagib pravca (β_1). Ukoliko malo promijenimo nagib, tada se i odsječak na osi y jako mijenja jer se podaci nalaze daleko od središta koordinatnog sustava. Drugim riječima, parametri su jako korelirani. Tada će MCMC algoritmi teško pretražiti skup stanja. Na primjer, Gibbsov će algoritam stalno praviti jako male korake jer on radi samo vodoravne ili okomite korake, a masa aposteriori distribucije je koncentrirana oko dijagonale kod koreliranih parametara. Taj problem možemo riješiti standardizacijom podataka. Ukoliko se podaci nalaze blizu ishodišta koordinatnog sustava, tada mala promjena nagiba neće jako utjecati na promjenu odsječka na osi y. Dakle, podatke moramo standardizirati. Kada ih standardiziramo, može se pokazati da koeficijenti regresije moraju pasti u segment $[-1, 1]$ u regresiji najmanjih kvadrata. Zato ćemo za apriori distribuciju od parametara $\beta_0, \beta_1, \beta_2$ uzeti normalnu razdiobu s očekivanjem 0 i standardnom devijacijom 2, tj.

$$\beta_j \sim N(0, 2), \quad j = 1, 2, 3.$$

Gustoć ove razdiobe je prilično fiksna na vrijednostima između -1 i 1. Nakon što standardiziramo podatke i odredimo standardizirane parametre, njih ćemo pretvoriti u nestandardizirane parametre kako bismo ih mogli koristiti na nestandardiziranim podacima i radi lakše interpretacije.

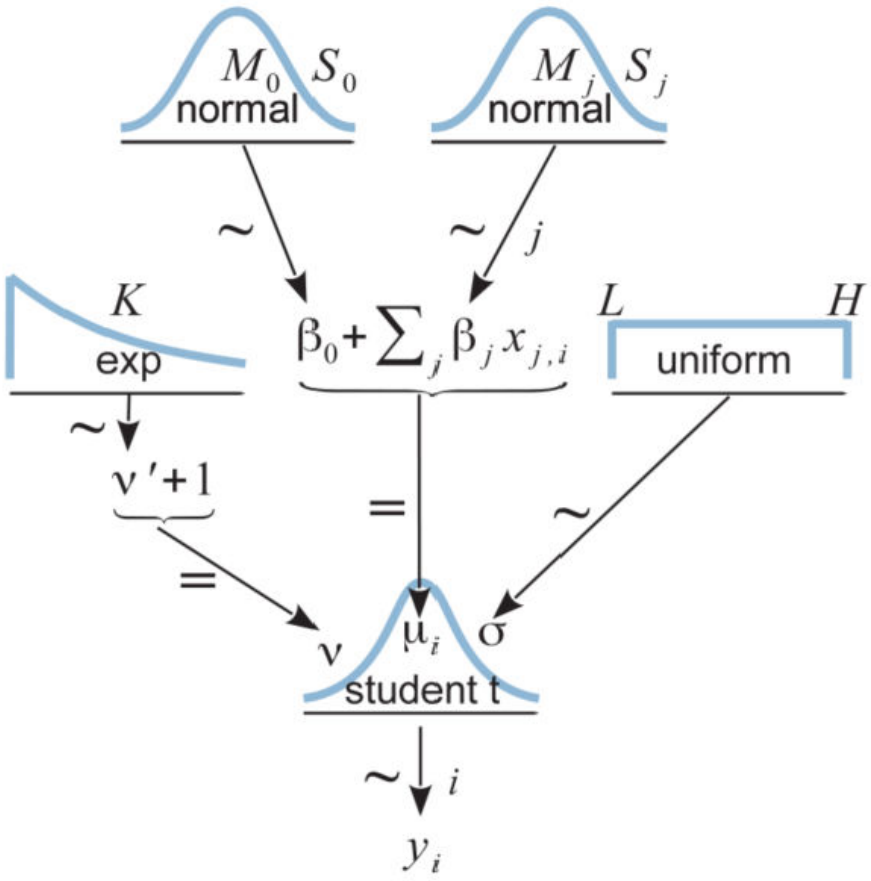
Vidimo da između parametara imamo hijerarhijsku strukturu. U bayesovskoj statistici česti su hijerarhijski modeli, a ovo je jedan primjer. Vrlo ih je jednostavno i korisno prikazivati slikom. Prikaz gore opisanog modela može se vidjeti na slici 7. Slika je preuzeta iz knjige [3]. Ovakve slike su posebno korisne i kod implementacije modela u JAGS-u jer svaka strelica predstavlja jednu liniju koda u JAGS-u, no o tome više u Dodatku.

Nakon što smo definirali model i apriori distribucije, trebamo primijeniti Bayesovo pravilo i izračunati aposteriori distribuciju. Ukoliko sa D označimo podatke, tada trebamo izračunati:

$$p(\nu, \sigma, \beta_0, \beta_1, \beta_2 | D) = \frac{p(D|\nu, \sigma, \beta_0, \beta_1, \beta_2)\, p(\nu, \sigma, \beta_0, \beta_1, \beta_2)}{\int\int\int\int\int p(D|\nu, \sigma, \beta_0, \beta_1, \beta_2)\, p(\nu, \sigma, \beta_0, \beta_1, \beta_2)\, d\nu\, d\sigma\, d\beta_0\, d\beta_1\, d\beta_2}. \tag{6}$$

Zbog nezavisnosti parametara u odnosu na apriori razdobu vrijedi:

$$p(\nu, \sigma, \beta_0, \beta_1, \beta_2) = p(\nu)p(\sigma)p(\beta_0)p(\beta_1)p(\beta_2).$$



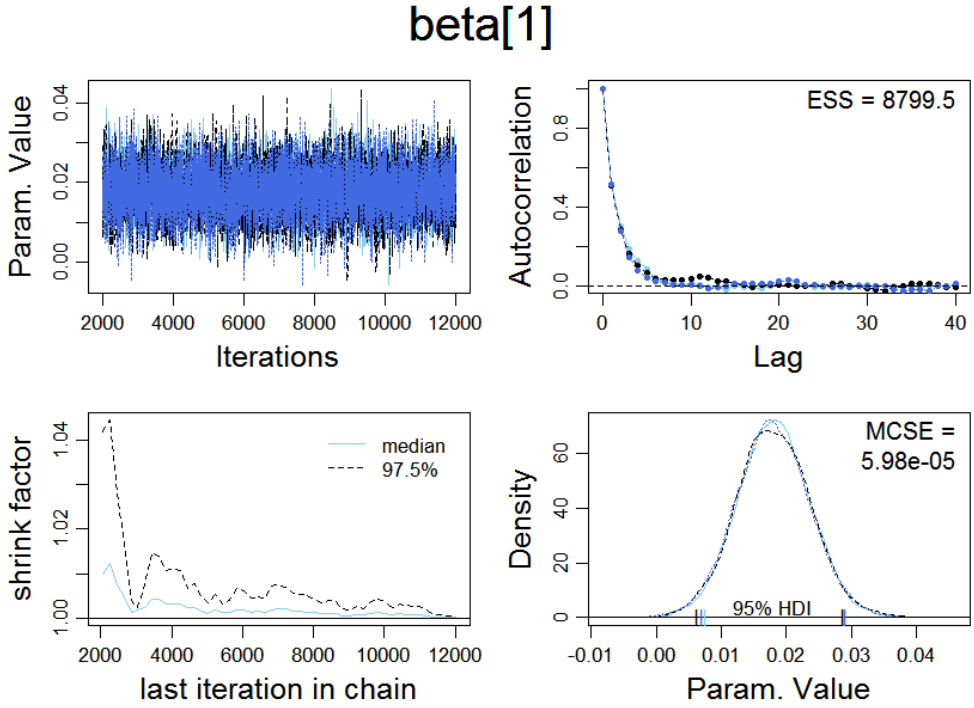
Slika 7: Hijerarhijski model robusne linearne regresije s pripadnim apriori distribucijama parametara

Dakle, integral u nazivniku je umnožak funkcije vjerodostojnosti, koja je umnožak funkcija gustoća t-razdiobe, i apriori gustoće, koja je umnožak gustoća eksponencijalne distribucije, normalne distribucije i uniformne distribucije. Takav integral vrlo je teško izračunati ili aproksimirati numeričkim metodama. Upravo ovdje su nam korisni MCMC algoritmi koji će nam omogućiti da dođemo do velikog uzorka iz aposteriori distribucije. Bez njih bi ovakva analiza stala na ovom mjestu. Upravo zato je bayesovska statistika doživjela porast upravo kada su računala postala dovoljno jaka da uz pomoć MCMC algoritama možemo dobiti vjerodostojnu reprezentaciju aposteriori distribucije. Nakon što dobijemo uzorak, preostalo nam je samo provjeriti predstavlja li dobiveni uzorak aposteriori distribuciju i, ako da, proučiti ju.

Dijagnostika

Nakon što smo generirali uzorak uz pomoć MCMC algoritama, moramo provjeriti reprezentira li taj uzorak aposteriori distribuciju. Lanac mora pretražiti cijeli prostor stanja, a rezultati ne smiju ovisiti o početnom stanju iz kojeg lanac kreće.

Osim reprezentativnosti, treba obratiti pažnju i na preciznost i stabilnost procjena te efikasnost.



Slika 8: Dijagnostika za parametar β_1

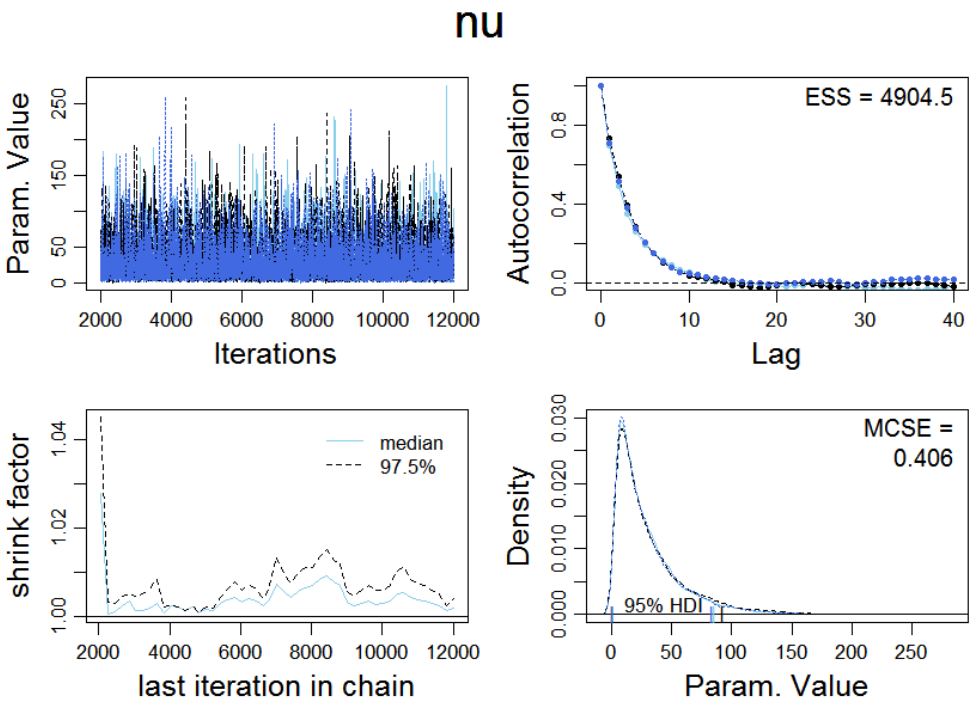
Jednostavan način provjeravanja reprezentativnosti je pokrenuti više lanaca i nacrtati njihova kretanja za svaki parametar. Ako lanci dobro reprezentiraju aposteriori distribuciju, onda će često posjećivati približno iste vrijednosti. Također, za svaki lanac možemo nacrtati izglađen histogram, tj. procjenjenu gustoću (eng. *density plot*). Te gustoće trebale bi biti što sličnije jer svaki lanac treba predstavljati istu aposteriori distribuciju. Postoje i numerički pokazatelji kao što je Gelman-Rubinova statistika koja se često na engleskom zove i *shrink factor*. Ona mjeri varijabilnost između lanaca i varijabilnost unutar lanca. Ukoliko lanci vjerno reprezentiraju aposteriori distribuciju, onda će te varijabilnosti biti jednake, tj. Gelman-Rubinova statistika bit će blizu 1. Ako je neki lanac zapeo i ne reprezentira aposteriori distribuciju vjerno, onda će se varijabilnost između lanaca povećati u odnosu na varijabilnost unutar lanca. Heuristika kaže da ukoliko je Gelman-Rubinova statistika iznad 1.1, onda se trebamo zabrinuti da lanci možda nisu dovoljno iskonvergirali.

Mjere kojima provjeravamo preciznost već smo spomenuli u poglavlju 2.4. To su efektivna veličina uzorka (ESS) i Monte Carlo standardna pogreška.

Što se tiče efikasnosti, napomenimo samo da pokretanje više lanaca na računalu možemo izvršiti paralelno jer oni ne ovise jedni o drugima. Sve njihove putanje trebamo koristiti kao uzorak iz aposteriori distribucije ukoliko su reprezentativni.

U našem primjeru pokrenuli smo 3 lanca s 1 000 (*burn-in*) koraka i 10 000 koraka nakon toga. Za svaki parametar izračunali smo gore navedene statistike. Rezultate

smo prikazali na slikama 8 i 9 za parametre β_1 i ν.



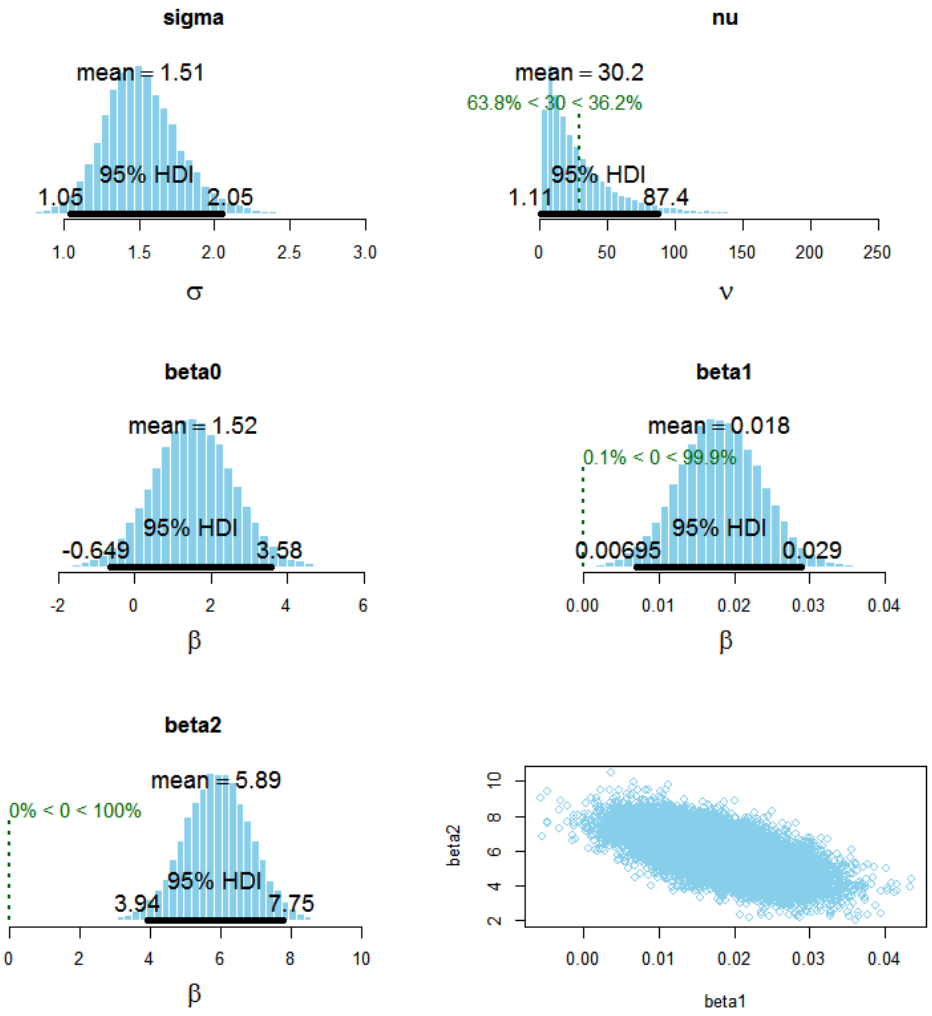
Slika 9: Dijagnostika za parametar ν

Na slici 8 gore lijevo vidimo putanje tri pokrenuta lanca. Putanje idu od 2 000 do 12 000 iteracija jer smo 1 000 koraka koristili kao adaptacijske korake (u kojima JAGS prilagođava MCMC algoritam, npr. skalira skokove u Metropolis-Hastings algoritmu), a drugih 1 000 kao *burn-in* period. Vidimo da lanci često posjećuju približno iste vrijednosti. Na slici gore desno vidimo da je efektivna veličina uzorka dosta velika i da autokorelacija lanaca brzo pada povećanjem odmaka. Na slici dolje lijevo vidimo da je Gelman-Rubinova statistika cijelo vrijeme ispod 1.1. Na slici dolje desno vidimo da su dobivene aposteriori distribucije jako slične te smo izračunali MCSE. Iz svega ovoga zaključujemo da lanci dobro i precizno reprezentiraju dobivenu aposteriori distribuciju i da možemo koristiti dobiveni uzorak za daljnju analizu. Slične su slike i za sve ostale parametre, a kao primjer dajemo još i sliku 9 za parametar ν.

Napomenimo još da postoje autori koji zagovaraju drugačije pristupe za dijagnostiku MCMC algoritama. Na primjer, govore da nije potrebno izbacivati početne korake lanca (*burn-in* period) i da je bolje imati jedan dugačak lanac nego više kraćih.

Proučavanje aposteriori distribucije

Nakon što smo provjerili da generirani MCMC uzorak vjerno reprezentira aposteriori distribuciju, prelazimo na proučavanje dobivenih rezultata. Za svaki parametar crtamo histogram marginalne aposteriori distribucije, 95% HDI i neku mjeru centralne tendencije. Mi smo nacrtali prosjek. Često je od značaja i mod distribucija kao

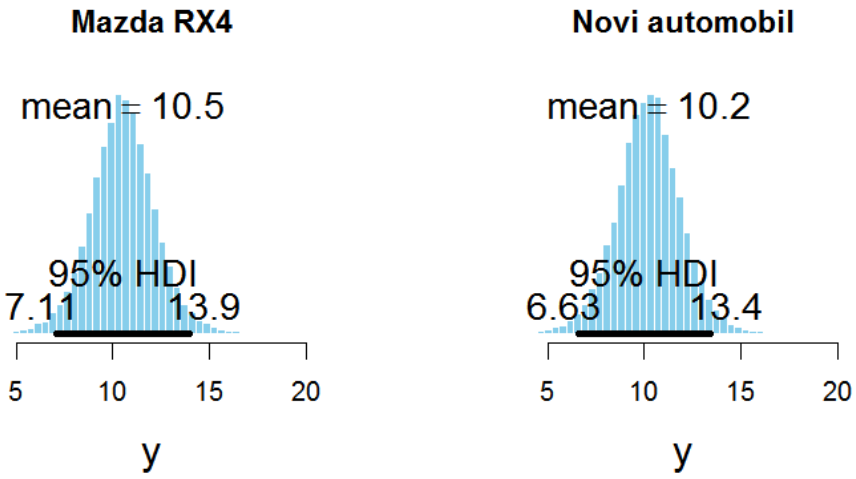


Slika 10: Histogrami aposteriori distribucije za svaki parametar

točka s najvećom gustoćom. Rezultate možemo vidjeti na slici 10. Najviše nas zanimaju parametri β_1 i β_2 jer oni predstavljaju informacije o tome kako broj konjskih snaga i težina automobila utječu na njegovu potrošnju. Ukoliko kao procijene vrijednosti parametara uzmemo očekivanje marginalne distribucije, možemo zaključiti da automobil koji ima 100 konjskih snaga više, u prosjeku troši 1.8 više litara na 100 kilometara. Automobil koji je 100 kila teži, troši prosječno 0.59 litara više na 100 kilometara. Uočimo da to što parametar β_1 poprima vrlo male vrijednosti, ne znači da broj konjskih snaga malo utječe na potrošnju. Veličine parametara β_1 i β_2 ovise o skali. Broj konjskih snaga uglavnom je troznamenkasti broj dok je težina jednoznamenkasti broj. Iz ove analize zaključujemo da obje varijable svojim povećanjem povećavaju potrošnju goriva automobila.

Još jedna posljedica korištenja bayesovske statistike je ta što ukoliko za neku konkretnu kombinaciju zavisnih varijabli želimo dobiti procjenu vrijednosti ovisne

varijable, možemo dobiti cijelu distribuciju ovisne varijable. Samim time imamo više informacija od obične točkovne procjene i lagano možemo izračunati pouzdane intervale. Da bismo to izračunali, u svakom koraku MCMC algoritma za trenutnu kombinaciju parametara generiramo slučajnu vrijednost iz razdiobe Y_i. U našem slučaju, u svakom koraku za trenutne vrijednosti parametara ν, σ, β_0, β_1 i β_2 generiramo vrijednost iz $t(\nu; \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}, \sigma)$.



Slika 11: Razdioba potrošnje goriva za automobil sa 110 konjskih snaga i težinom 1.18 tona (Mazda RX4, lijevo) i za automobil s 90 konjskih snaga i težinom 1.2 tone (desno)

Na slici 11 možemo vidjeti distribuciju vrijednosti potrošnje goriva za Mazdu RX4, automobil iz naših podataka čija je izmjerena potrošnja 11.2 litara na 100 kilometara. Također, možemo zamisliti situaciju gdje proizvodimo automobil i zanima nas njegova potrošnja, a znamo koliko će imati konjskih snaga i kolika će mu biti težina. Na slici smo nacrtali i 95% interval najveće gustoće.

Dodatak

Uvod u JAGS

U ovom dodatku dat ćemo kratki uvod u program JAGS. Prvo ćemo objasniti kako se koristi kroz programski jezik R na jednostavnom primjeru gdje imamo jednodimenzionalne podatke iz normalne distribucije. Zatim ćemo pokazati kako smo koristili JAGS na primjeru potrošnje automobila. Podrazumijeva se osnovno poznavanje programskog jezika R.

JAGS je program koji automatski gradi MCMC algoritme za kompleksne hijerarhijske modele. Prva verzija napravljena je krajem 2007. godine dok je zadnja stabilna verzija izašla u siječnju 2016. godine. Autor programa je Martyn Plummer. Program je besplatan. JAGS je skraćenica engleskog naziva *Just another Gibbs sampler*. On je jedan od programa kao što su Stan ili BUGS koji se koriste za primjenu bayesovske statistike i gradnju hijerarhijskih modela. Korištenje JAGS-a može se lagano objasniti: on uzima korisnikov opis hijerarhijskog modela podataka i vraća MCMC uzorak iz aposteriori distribucije. Mi ćemo JAGS koristiti kroz jezik R uz pomoć R-ovih paketa `rjags` i `runjags`.

Prvo ćemo učitati potrebne pakete za komunikaciju R-a i JAGS-a te generirati podatke iz normalne razdiobe s očekivanjem 50 i standardnom devijacijom 11. Te podatke trebamo poslati JAGS-u kroz listu u R-u.

```
library(runjags)
library(rjags)

set.seed(1912)
N = 100
y = rnorm(N,mean = 50,sd = 11)

dataList = list( y = y, N = N )
```

Pridruživanje `y = y` u listi može izgledati neobično, no zapravo desni `y` predstavlja varijablu u R-u koja sadrži vektor naših podataka, a `y` s lijeve strane naziv te varijable u listi. Nakon što smo pripremili podatke, moramo definirati model. Pretpostavljamo da naši podaci dolaze iz normalne razdiobe s očekivanjem μ i standardnom devijacijom σ koje želimo procijeniti, tj.

$$Y_i \sim N(\mu, \sigma^2).$$

Zatim moramo odabrati apriori razdiobe za parametre μ i σ. Uzmimo sljedeće apriori distribucije:

$$\mu \sim N(0, 1000^2) \quad \text{i} \quad \sigma \sim U(10^{-5}, 10^5).$$

Imamo tri pridruživanja što će odgovarati trima linijama koda u JAGS-u. Model ćemo prvo zapisati u *string*.

```
modelString = "
model
{
    for( i in 1:N )
    {
        y[i] ~ dnorm(mu, 1/sigma^2) #vjerodostojnost
    }
    mu ~ dnorm(0, 1/1000^2) #apriori razdioba
    sigma ~ dunif( 10^(-5), 10000) #apriori razdioba
}

writeLines(modelString, con = "TEMPmodel.txt")
"
```

Unutar dvostrukih navodnika počinje JAGS kod. Model definiramo ključnom riječi `model` i otvaramo vitičaste zagrade. U JAGS modelu možemo koristiti samo varijable koje smo poslali u listi i to točno tim nazivom. Dakle, smijemo koristiti `N` koji predstavlja veličinu uzorka i `y` koji predstavlja vektor podataka. Prvo ćemo reći da svaki podatak dolazi iz normalne razdiobe s očekivanjem μ i varijancom σ^2. To radimo pomoću `for` petlje za svaki podatak koja ima sintaksu kao u R-u. Zatim simbolom `~` govorimo kako je distribuirana ta slučajna varijabla, a s `dnorm` govorimo da se radi o gustoći normalne distribucije. Kod američkih oznaka za normalnu distribuciju koristi se očekivanje i preciznost τ. Preciznost i varijanca povezani su preko formule:

$$\tau = \frac{1}{\sigma^2}.$$

Zato kao drugi argument od `dnorm` stavljamo $1/\sigma^2$. Nakon toga, na isti način pridružujemo apriori razdiobe parametrima μ i σ. Točan poredak svih tih pridruživanja JAGS-u nije bitan. Nakon što smo u *string* spremili opis modela, taj *string* zapisujemo u datoteku pomoću naredbe `writeLines`. Do sada zapravo još nismo ništa radili unutar JAGS-a. Sada ćemo poslati podatke i model JAGS-u te će nam on vratiti objekt koji predstavlja sagrađeni model.

```
jagsModel = jags.model( file = "TEMPmodel.txt",
    data = dataList, n.chains = 3, n.adapt = 500)
```

Funkcija `jags.model` prima dokument u kojem se nalazi opis modela i podatke spremljene u listu. Zatim možemo zadati koliko lanaca želimo pokretati i koliko adaptacijskih koraka koristimo. Adaptacijski koraci služe JAGS-u da odabere najbolji algoritam i da ga što bolje prilagodi - na primjer, da skalira Metropolis-Hastings algoritam kako bi dao najbolje rezultate. Nakon toga naredbom `update` govorimo koliko koraka želimo da svaki lanac prođe kao *burn-in* period. Kao argumente prima objekt koji predstavlja JAGS model i broj koraka ili iteracija.

```
update(jagsModel, n.iter = 1000)
```

Na kraju pomoću funkcije `coda.samples` dobivamo uzorak iz aposteriori distribucije. U funkciju šaljemo JAGS model i imena varijabli koje želimo zapamtiti te broj koraka koje će svaki lanac prijeći.

```
codaSamples = coda.samples(jagsModel,
    variable.names = c("mu","sigma"),
    n.iter = 3333)
```

Sada ukoliko izvršimo naredbu

```
uzorak = as.matrix(codaSamples)
```

imamo matricu u R-u od 9 999 redaka i 2 stupca koja predstavlja uzorak iz aposteriori distribucije. Sada lagano možemo nacrtati histogram marginalnih distribucija, izračunati medijan, prosjek i sve što nas zanima o marginalnim aposteriori distribucijama. No, kako se ne bismo mučili s time, koristit ćemo gotove funkcije koje je napravio J. Kruschke i podijelio sa svim zainteresiranima, a nalaze se u datoteci `DBDA2E-utilities.R` i moramo ih učitati u R. Mi smo tu datoteku stavili u radni direktorij i učitali je pomoću naredbe `source`.

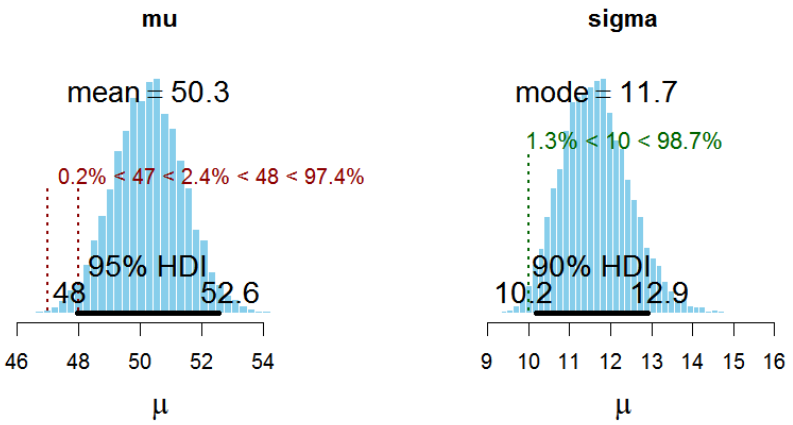
```
source("DBDA2E-utilities.R")

diagMCMC( codaObject = codaSamples, parName = "mu" )
diagMCMC( codaObject = codaSamples, parName = "sigma" )

par(mfrow = c(1,2))
plotPost( codaSamples[,"mu"], main = "mu",
    xlab = bquote(mu), cenTend = "mean",  ROPE = c(47,48))

plotPost( codaSamples[,"sigma"], main = "sigma",
    xlab = bquote(mu), cenTend = "mode",
    credMass = 0.9, compVal = 10)
```

Naredba `diagMCMC` radi dijagnostiku lanaca za parametar kako bismo provjerili reprezentativnost i preciznost dobivenog uzorka. Naredba `plotPost` crta histogram aposteriori distribucije i interval najveće gustoće (HDI). Možemo izabrati koju mjeru centralne tendencije želimo nacrtati (`mode, median, mean`) i koliki HDI interval želimo (`credMass = 0.9`). Postoje i opcije za uspoređivanje s nekom vrijednosti ili nekim intervalom koje se koriste kod donošenja binarnih odluka o vrijednosti parametra, a mi smo ih naveli radi ilustracije. Dobivene slike mogu se vidjeti na slici 12.

Primjer: Linearna regresija s teškim repovima

Podaci o potrošnji automobila i njihovih karakteristika ugrađeni su u R-ovu biblioteku `datasets` pod varijablom `mtcars`. Svatko tko ima programski jezik R, može lagano instalirati tu biblioteku i doći do podataka. Podaci su izvučeni iz automobilskog magazina *Motor Trend US* iz 1974. godine.

Cijeli proces provodimo u R-u kao i u prethodnom primjeru.



Slika 12: Marginalna aposteriori distribucija za parametre μ i σ

```
library(datasets)
library(rjags)
library(runjags)
library(metRology)
source("DBDA2E-utilities.R")

...

dataList = list(y=y, x=x, n = n, m = m)
```

Paket `metRology` koristimo radi generalizirane t-distribucije. U varijablu `y` spremili smo vektor potrošnje automobila na 100 kilometara dok smo u varijablu `x` spremili *dataframe* koji sadrži konjske snage i težinu za svaki automobil. Varijabla `m` predstavlja broj automobila, a varijabla `n` broj neovisnih varijabli.

```
> head(x)
                   hp       wt
Mazda RX4         110 1.188406
Mazda RX4 Wag     110 1.304071
Datsun 710         93 1.052329
Hornet 4 Drive    110 1.458292
Hornet Sportabout 175 1.560350
Valiant           105 1.569421
```

Model smo definirali i objasnili u poglavlju 3.3, a sada dajemo kod tog modela u JAGS-u. Rekli smo da ćemo podatke prvo standardizirati. To radimo tako da svakoj varijabli oduzimamo aritmetičku sredinu i dijelimo standardnom devijacijom. Taj proces ćemo također napraviti u JAGS-u pomoću `data` bloka. Zatim slijedi opis modela koji izgleda kompliciranije jer radimo sa standardiziranim podacima koje na kraju vraćamo u nestandardizirani oblik. Kada je neka varijabla standardizirana,

ispred nje u kodu pišemo slovo z.

```
modelString = "
data
{
    ym = mean(y)
    ysd = sd(y)
    for( i in 1:n )
    {
        zy[i] = (y[i] - ym)/ysd
    }

    for( j in 1:m )
    {
        xm[j] = mean(x[,j])
        xsd[j] = sd(x[,j])
        for( i in 1:n )
        {
            zx[i,j] = (x[i,j] - xm[j])/xsd[j]
        }
    }
}

model
{
    for( i in 1:n )
    {
        zy[i] ~ dt( zbeta0 + sum( zbeta[1:m] * zx[i,1:m] ),
                    1/zsigma^2, nu ) #vjerodostojnost
    }
    zbeta0 ~ dnorm(0, 1/2^2) #apriori distribucija
    for( j in 1:m )
    {
        zbeta[j] ~ dnorm(0, 1/2^2) #apriori distribucija
    }
    zsigma ~ dunif( 10^(-5), 10000) #apriori distribucija
    nu = nuMinusOne + 1
    nuMinusOne ~ dexp(1/29.0) #apriori distribucija

    beta[1:m] = ( zbeta[1:m] / xsd[1:m])* ysd
    beta0 = zbeta0*ysd + ym -
          sum( zbeta[1:m] * xm[1:m] / xsd[1:m] ) * ysd
    sigma = zsigma*ysd
}
```

”

Nakon `data` bloka koji ima sintaksu kao R-ov kod, implementiramo hijerarhijski model linearne regresije s teškim repovima. Nakon toga vrijednosti vraćamo u nestandardizirani oblik. Daljnji postupak jednak je kao u početnom primjeru. Spremamo string u datoteku, radimo objekt koji sadrži JAGS model i dobivamo uzorak iz aposteriori distribucije.

```
writeLines(modelString, con = "TEMPmodel.txt")

jagsModel = jags.model(file = "TEMPmodel.txt",
    data = dataList, n.chains = 3,
   n.adapt = 1000)

update(jagsModel, n.iter = 1000)

codaSamples = coda.samples(jagsModel,
   variable.names = c("beta0","beta","nu","sigma"),
   n.iter = 10000)
```

Na kraju smo opet koristili funkcije `diagMCMC` i `plotPost` kako bismo provjerili i interpretirali dobivene rezultate.

Bibliografija

[1] J. M. Flegal, M. Haran i G. L. Jones. Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*, pages 250–260, 2008.

[2] R. E. Kass, B. P. Carlin, A. Gelman i R. M. Neal. Markov chain Monte Carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

[3] J. Kruschke. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press, 2014.

[4] A. Lisičar. *Markovljevi lanci na općenitom skupu stanja*. 2014.

[5] S. P. Meyn i R. L. Tweedie. *Markov chains and stochastic stability*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993.

[6] C. P. Robert i G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer, 11. 2010.

[7] N. Sarapa. *Teorija vjerojatnosti*. Udžbenici Sveučilišta u Zagrebu. Školska knjiga, 2002.

Sažetak

U ovom radu govorimo o MCMC algoritmima i ilustriramo njihovu primjenu u bayesovskoj statistici. MCMC algoritmi služe za simuliranje uzorka iz distribucije s gustoćom f koju moramo znati samo do na konstantu. Oni konstruiraju Markovljev lanac čija se putanja uzima kao uzorak iz distribucije s gustoćom f, što nam omogućuje ergodski teorem.

Nakon motivacijskog uvoda, u prvom poglavlju obrađujemo nužnu teoriju koja je u pozadini MCMC algoritama, a to je teorija Markovljevih lanaca na općenitom skupu stanja. Prvo definiramo osnovne pojmove poput prijelazne jezgre i Markovljevog lanca, a zatim neka svojstva Markovljevih lanaca poput ireducibilnosti, povratnosti, Harrisove povratnosti i stacionarnu distribuciju Markovljevog lanca. Iskazujemo teorem koji nam daje dovoljne uvjete da marginalne distribucije lanca konvergiraju prema stacionarnoj distribuciji f i ergodski teorem koji nam omogućuje da putanju lanca koristimo kao uzorak dobiven iz stacionarne distribucije f.

Nakon toga obrađujemo Metropolis-Hastings algoritam, jedan od najvažnijih MCMC algoritama. Dokazujemo da putanju lanca dobivenog ovim algoritmom možemo koristiti kao uzorak iz zadane ciljne distribucije f. Uzorak iz distribucije f služi nam kako bismo procijenili karakteristike distribucije f pa smo uveli efektivnu veličinu uzorka i Monte Carlo standardnu grešku kao mjere preciznosti dobivenih procjena. Kao primjere Metropolis-Hastings algoritma obradili smo Nezavisni Metropolis-Hastings algoritam i Metropolis-Hastings algoritam sa slučajnom šetnjom. Na primjeru smo vidjeli da je kod Nezavisnog Metropolis-Hastings algoritma najbolje odabrati *proposal* distribuciju što sličniju ciljnoj distribuciji, ukoliko je to moguće. Kod Metropolis-Hastings algoritma sa slučajnom šetnjom uočili smo da za efikasnost algoritma moramo odgovarajuće skalirati *proposal* distribuciju. Na kraju smo spomenuli i Gibbsovo uzorkovanje koji je vrlo važan i u primjenama čest algoritam.

U zadnjem dijelu ovoga rada objašnjavamo glave ideje bayesovske statistike - pojmove poput apriori i aposteriori distribucije. Na primjeru bacanja novčića ilustriramo kako se aposteriori distribucija ponekad može izračunati egzaktno, no uočavamo i da se to ponekad neće moći zbog čega onda koristimo MCMC algoritme. Nakon toga ukratko objašnjavamo generalizirane linearne modele kako bismo mogli napraviti model linearne regresije s teškim repovima na bayesovski način. Kao primjer uzeli smo određivanje potrošnje goriva automobila. Nakon definiranja modela komentiramo izbor apriori distribucija te koristimo MCMC algoritme kako bismo dobili uzorak iz aposteriori distribucije. Prije nego što analiziramo rezultate, radimo dijagnostiku kako bismo provjerili da dobiveni uzorak doista predstavlja uzorak iz aposteriori distribucije. U dodatku objašnjavamo kako smo tehnički proveli ranije opisan postupak pomoću programskog jezika R i programa JAGS.

Summary

In this paper we talk about Markov chains Monte Carlo (MCMC) algorithms, and we ilustrate their application in Bayesian statistics. MCMC algorithms are used to generate sample from distribution with density f, which is known up to a normalising constant only. They produce Markov chains whose trajectory represents a sample from distribution with the density f, which is justified by Ergodic theorem.

After motivational introduction, in chapter one, we introduce necessary background theory of MCMC algorithms; that is the theory of Markov chains on general state spaces. First, we define basic terms such as transition kernels and Markov chains, and after that we define some properties of Markov chains such as irreducibility, recurrence, Harris recurrence and stationary distribution of Markov chain. We state a theorem which gives us necessary conditions on the chain, so that its marginal distributions converge to its stationary distribution f, and Ergodic theorem.

Secondly, we define Metropolis-Hastings algorithm, one of the most important MCMC algorithms. We prove that we can use a trajectory generated with Metropolis-Hastings algorithm as sample from target distribution f. Sample from distribution f is used to approximate distribution f and its characteristics, so we introduced effective sample size and Monte Carlo standard error as measures of accuracy. Furthermore, we give examples of Metropolis-Hastings algorithm: the Independent Metropolis-Hastings and Random walk Metropolis-Hastings. On a simple example we demonstrate that in the Independent Metropolis-Hastings algorithm optimal proposal distribution should be as much proportional to target distribution as possible. In the Random walk Metropolis-Hastings algorithm we noticed that we need to scale proposal distribution for effectiveness. Finally, we mention Gibbs sampler which is very important and widely used algorithm.

In the last part of this paper, we explain the main ideas of Bayesian statistics. We introduce terms like prior and posterior distribution. On a simple coin tossing example we illustrate that sometimes one can exactly calculate posterior distribution, but we notice that in cases when that is not possible, one can use MCMC algorithms. After that, we briefly explain generalised linear models, so that we can introduce heavy-tailed linear regression model from Bayesian perspective. As an example we are estimating car fuel consumption. After defining the model, we comment on choosing prior distributions, and we use MCMC algorithms to get sample from posterior distribution. Before we analize posterior distribution, we check if our sample is representative of prior distribution and accurate enough. In the appendix, we explain how we used programming language R and JAGS to give bayesian analysis of the heavy-tailed linear regression model.

Životopis

Moje ime je Elio Bartoš i rođen sam 29. siječnja 1993. godine u Bjelovaru. Odrastao sam u gradu Daruvaru gdje sam završio Osnovnu školu Vladimira Nazora. Aktivno sam se bavio sportom. Trenirao sam nogomet i stolni tenis te sudjelovao na lokalnim natjecanjima. Godine 2011. završio sam srednju školu Gimnazija Daruvar i upisao sam preddiplomski studij matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu. Preddiplomski studij završio sam 2014. godine nakon kojega sam upisao diplomski studij Matematičke statistike na istoimenom fakultetu.

Tijekom diplomskog studija sudjelovao sam na studentskom natjecanju Mozgalo, natjecanju u analiziranju i obrađivanju podataka. Na ljeto 2015. godine odradio sam studentsku praksu u Hrvatskoj narodnoj banci u sektoru za financijsku stabilnost.