

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku

Alen Kovačević
Monte Carlo simulacije
Diplomski rad

Osijek, 2015. godina

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku

Alen Kovačević
Monte Carlo simulacije
Diplomski rad

Mentor: doc. dr. sc. Nenad Šuvak

Osijek, 2015. godina

Sadržaj

1	Uvod	4
2	Generiranje slučajnih brojeva	5
2.1	Uniformno distribuirani slučajni brojevi	5
2.2	Linearni kongruentni generatori	9
2.3	Nelinearni kongruentni generatori	12
3	Generiranje slučajnih varijabli	13
3.1	Metoda inverzne transformacije	15
3.2	Metoda prihvatanja-odbijanja	18
4	Monte Carlo metode	22
4.1	Procjena integrala	23
4.2	Eksperimentalna greška u Monte Carlo metodama	25
4.3	Varijanca Monte Carlo procjenitelja	26
5	Markov Chain Monte Carlo	30
5.1	Markovljevi lanci	30
5.2	Metropolis-Hastings algoritam	37
5.2.1	Primjeri	39
	Literatura	45
	Sažetak	46
	Title and summary	47
	Životopis	48

1 Uvod

Monte Carlo metode, odnosno simulacije, široka su klasa algoritama koji se oslanjaju na opetovano slučajno uzorkovanje u svrhu dobivanja numeričkih rezultata. Uloga Monte Carlo metoda u svim granama znanosti zadnjih nekoliko godina sve više raste. Često se koriste u matematičkim i fizikalnim problemima, a najpogodnije su za uporabu u situacijama kada je teško ili nemoguće dobiti procjenu nekog izraza u konačnom broju koraka, kao i u slučajevima kada je primjena determinističkog algoritma neizvediva. Tehnika je to koja se koristi u posve različitim područjima, poput financija, proizvodnje, energetike, osiguranja i mnogih drugih.

Porast snage računala i razvoj metodologije simulacija, doveli su do prepoznavanja izračuna koji koriste algoritme ili protokole kao trećeg pristupa za napredak prirodnih znanosti, zajedno s poznavanjem teorije i tradicionalnim eksperimentiranjem. U jezgri Monte Carlo simulacija nalazi se generiranje slučajnih brojeva. Mnoge aplikacije, uključujući i Monte Carlo metode, umjesto slučajnih brojeva koriste *pseudoslučajne* brojeve, koji su deterministički ali "izgledaju" kao da su generirani slučajno.

U radu smo najprije predstavili razne generatore slučajnih brojeva i pružili uvid u svojstva o kojima treba voditi računa prilikom njihova korištenja ili konstruiranja. Zatim se bavimo generiranjem slučajnih varijabli, gdje smo najviše pozornosti posvetili metodi *inverzne transformacije* i metodi *prihvatanja-odbijanja*. Nakon toga prešli smo na Monte Carlo metode, odnosno upoznavanje s najopćenitijim oblikom metode, prilikom čega smo razmatrali određene rezultate i probleme. Zadnje poglavlje analizira Markov Chain Monte Carlo, gdje se, nakon što se podsjetimo osnovnih pojmova iz područja Markovljevih lanaca, upoznajemo s najkorištenijim *Metropolis-Hastings* algoritmom. Na kraju, primjere obrađenih algoritama predstavljamo kroz implementaciju u programskom jeziku R.

2 Generiranje slučajnih brojeva

Kako se brojne statističke metode oslanjaju na slučajne uzorke, javlja se potreba za izvorom slučajnih brojeva. Dok su se prije koristile tablice slučajnih brojeva, razvojem računala one su zamijenjene generatorima slučajnih brojeva. U začetima simuliranja, do slučajnosti se dolazilo ručnim tehnikama, poput bacanja novčića i igraće kockice. Kasnije, mehanički uređaji spajani su na računala jer je prevladavalo mišljenje kako samo mehanički ili elektronički uređaji mogu proizvesti uistinu slučajne nizove. No, budući da su mehaničke metode prespore za široku uporabu te uz činjenicu da nizovi ne mogu biti reproducirani i otkriće da su generirani brojevi zavisni, ova tehnika ubrzo je odbačena. Velika većina današnjih generatora slučajnih brojeva bazirana je na algoritmima koji se jednostavno mogu implementirati u računala. Dobar generator obuhvatit će sva važna statistička svojstva istinski slučajnog niza, unatoč tome što je on generiran determinističkim algoritmom. U ovom slučaju ipak se ne radi o brojevima koji su istinski slučajni, nego o brojevima koji se na prvi pogled takvima čine, ali način njihovog nastanka nije slučajan. Tako nastale brojeve stoga zovemo *pseudoslučajnim brojevima*.

2.1 Uniformno distribuirani slučajni brojevi

U većini slučajeva želimo da generirani pseudoslučajni brojevi simuliraju uniformnu distribuciju na jediničnom intervalu $(0, 1)$. Za tu distribuciju koristimo oznaku $\mathcal{U}(0, 1)$. Općenito, koristimo notaciju $\mathcal{U}(a, b)$ za označavanje neprekidne uniformne distribucije na intervalu (a, b) . Ta je distribucija pogodna za uporabu budući da postoji mnogo jednostavnih tehnika za transformaciju uniformnog uzorka u uzorke iz drugih distribucija. U temelju Monte Carlo simulacija nalaze se metode za generiranje uniformno distribuiranih slučajnih varijabli i metode za prethodno spomenute transformacije.

Da bi slučajne brojeve mogli koristiti u svakodnevnim primjenama, oni moraju biti realizacije nezavisnih i jednako distribuiranih slučajnih varijabli, to jest za niz slučajnih varijabli U_1, U_2, \dots mora vrijediti:

(i) Svaka $U_i \sim \mathcal{U}(0, 1)$

(ii) Sve U_i su međusobno nezavisne.

Puno je važnije drugo svojstvo, koje implicira da su bilo koje dvije slučajne varijable nekorelirane te, još općenitije, da se vrijednost varijable U_i ne može pretpostaviti iz U_1, \dots, U_{i-1} .

Generator slučajnih brojeva proizvodi konačan niz brojeva u_1, u_2, \dots, u_K na jediničnom intervalu. Generirane vrijednosti dijelom ovise o parametrima zadanim od strane korisnika. Dobar generator je onaj koji zadovoljava uvjet da se mali segmenti (u odnosu na K) niza u_1, \dots, u_K teško razlikuju od realizacija nezavisnih uniformnih slučajnih varijabli. Učinkovit generator prema tome vraća vrijednosti koje su u skladu s navedenim svojstvima (i) i (ii). Za provjeru uniformnosti, možemo testirati hipotezu o jednakosti vjerojatnosti da se slučajni brojevi rasporede u podintervale jediničnog intervala jednake duljine. Ukoliko ne odbacimo takvu nul hipotezu, zapravo nismo odbacili niti hipotezu o uniformnosti.

Prethodna razmatranja promotrit ćemo na primjeru *linearnog kongruentnog generatora*¹, koji je rezultat ponavljanja sljedećeg modela:

$$x_{i+1} \equiv ax_i \pmod{m} \quad (2.1)$$

$$u_{i+1} = x_{i+1}/m, \quad (2.2)$$

gdje su a i m cjelobrojne konstante koje određuju generirane vrijednosti, uz danu početnu vrijednost x_0 , koja je cijeli broj između 1 i $m - 1$ a obično je posebno zadana od korisnika. Operacija $y \pmod{m}$ vraća cjelobrojni ostatak od y nakon dijeljenja s m . Drugim riječima,

$$y \pmod{m} = y - \lfloor y/m \rfloor m, \quad (2.3)$$

gdje $\lfloor x \rfloor$ označava najveći cijeli broj manji ili jednak x . Kako je rezultat operacije \pmod{m} uvijek cijeli broj između 0 i $m - 1$, vrijednosti u_i proizašle iz (2.1)-(2.2) uvijek su između 0 i $(m-1)/m$; posebno, nalaze se u jediničnom intervalu.

Zbog svoje jednostavnosti i učinkovitosti, LCG spada među najraširenije generatore korištene u praksi. Detaljno ćemo ih obraditi u potpoglavlju 2.2.

¹LCG - Linear Congruential Generator

LCG se zadaje izrazima

$$x_{i+1} = f(x_i), \quad u_{i+1} = g(x_{i+1}), \quad (2.4)$$

za određene funkcije f i g . Ako dopustimo da x_i budu vektori, tada praktički svi generatori slučajnih brojeva odgovaraju prethodnoj generalnoj formi.

Promotrimo niz (x_i) stvoren u (2.1) pomoću linearnog kongruentnog generatora s vrijednostima $a = 6$ i $m = 11$. Počevši od $x_0 = 1$, sljedeća vrijednost je $6 \bmod 11 = 6$, nakon čega dolazi $(6 \cdot 6) \bmod 11 = 3$. Početna vrijednost $x_0 = 1$ tako proizvodi niz

$$1, 6, 3, 7, 9, 10, 5, 8, 4, 2, 1, 6, \dots$$

Jednom kada se vrijednost ponovi, ponavlja se čitav niz. Zaista, budući da računalo može prikazati samo konačan broj vrijednosti, bilo koje ponavljanje oblika (2.4) će u konačnici dovesti do prijašnjeg x_i , a zatim ponoviti sve vrijednosti koje su slijedile nakon tog x_i . U prethodnom su se primjeru svih 10 različitih cijelih brojeva između 1 i $m - 1$ pojavili u nizu prije nego se se ponovila neka vrijednost. Ako ostavimo $m = 11$ ali uzmemo $a = 3$, početna vrijednost $x_0 = 1$ vraća

$$1, 3, 9, 5, 4, 1, \dots,$$

dok $x_0 = 2$ vraća

$$2, 6, 7, 10, 8, 2, \dots$$

Na taj način se moguće vrijednosti $\{1, 2, \dots, 10\}$ dijele na dva dijela. To znači da bez obzira na odabrani x_0 , faktor $a = 3$ proizvodi samo 5 različitih brojeva prije nego što se ponovi, dok faktor $a = 6$ proizvodi svih 10 različitih brojeva. Za LCG koji proizvodi svih $m - 1$ različitih vrijednosti prije nego što se ponovi kažemo da ima *puni period*. U praksi bismo htjeli generirati na desetke milijuna različitih vrijednosti prije nego nam se bilo koja ponovi. To ne možemo postići samo s odabiranjem velikog m , zbog mogućnosti da loš odabir parametara a i m može dovesti do kratkih ciklusa unutar vrijednosti $\{1, 2, \dots, m - 1\}$.

S time na umu, razmatramo sljedeće okolnosti prilikom konstruiranja generatora slučajnih brojeva:

- *Duljina perioda.* Kao što je već spomenuto, bilo koji generator slučajnih brojeva oblika (2.4) će se s vremenom ponoviti. Prednost dajemo generatorima s duljim periodom, to jest generatorima koji proizvode više različitih vrijednosti prije ponavljanja. Najdulji mogući period za LCG modula m je $m - 1$. Za LCG s punim periodom, razmaci između stvorenih vrijednosti u_i su širine $1/m$; stoga, veći m ima veću mogućnost da što preciznije aproksimira uniformnu distribuciju.
- *Reproduktivnost.* Jedan od nedostataka istinski slučajnog niza je taj da se ne može lako reproducirati. Često je važno imati mogućnost ponovnog pokretanja simulacije koristeći jednake parametre kao i prije, ili da se jednaki parametri koriste u dvije ili više simulacija. To je lako ostvarivo pomoću LCG ili bilo koje druge procedure koja ima općeniti oblik (2.4) s korištenjem iste početne vrijednosti x_0 .
- *Brzina.* Budući da generator slučajnih brojeva može biti korišten nekoliko tisuća puta prilikom jedne simulacije, važno je da bude brz. Na prvi pogled teško je zamisliti jednostavniji ili brži algoritam od LCG. Ranije se na brzini pokušavalo dobiti tako što se, na primjer, parametar m odabirao tako da bude potencija broja 2, prilikom čega bi se operacija mod m implementirala bez eksplicitnog dijeljenja. No, s obzirom na trenutnu brzinu računala, tako dobiveno ubrzanje ne bi opravdalo generator s lošijim distribucijskim svojstvima.
- *Prenosivost.* Algoritam za generiranje slučajnih brojeva treba proizvesti jednaki niz neovisno o računalnoj platformi. Želja za brzinom i duljinom perioda povremeno dovodi do implementacija koje ovise o specifičnim reprezentacijama brojeva pojedinog uređaja, dok se neke implementacije LCG oslanjaju na način na koji određeno računalo postupi s overflowom².
- *Slučajnost.* Najvažniji i najteže ostvariv uvjet. Postoje dva opća aspekta o kojima se vodi računa prilikom konstruiranja generatora s prividnom slučajnosti: teorijska svojstva i statistički testovi. Generatori s dobrim teorijskim svojstvima podvrgnuti su statističkom promatranju.

²Stanje prilikom kojeg je podatak veći od prostora u koji treba biti spremljen

2.2 Linearni kongruentni generatori

1948. godine američki matematičar D. H. Lehmer kao izvor slučajnih brojeva predložio je jednostavni linearni kongruentni generator. U tom generatoru, svaki broj određuje svog sljedbenika pomoću jednostavne linearne funkcije i operacije modulo. Iako ovaj generator ima ograničenu mogućnost proizvodnje dugačkog niza brojeva koji bi izgledali kao nezavisne realizacije uniformne slučajne varijable, čini osnovu za druge, adekvatnije generatore.

Oblik linearnog kongruentnog generatora je

$$x_i \equiv (ax_{i-1} + c) \pmod{m}, \quad 0 \leq x_i < m; \quad (2.5)$$

a se zove multiplikator, c prirast, a m modul generatora. Često se uzima da je $c = 0$, pa se u tom slučaju govori o *multiplikativnom kongruentnom generatoru*:

$$x_i \equiv ax_{i-1} \pmod{m}, \quad 0 < x_i < m. \quad (2.6)$$

Niz koji nastaje iz rekurzije (2.5) naziva se *Lehmerov niz*. Svaki x_i pretvara se u broj unutar jediničnog intervala $(0, 1)$ dijeljenjem s m , to jest,

$$u_i = x_i/m.$$

Ako su a i m dobro odabrani, vrijednosti u_i izgledat će kao da su slučajno i uniformno distribuirane između 0 i 1. Izraz (2.6) za cijele brojeve ekvivalentan je izrazu

$$u_i \equiv au_{i-1} \pmod{1}, \quad 0 < u_i < 1.$$

Budući da je x_i određen pomoću x_{i-1} i kako postoji samo m mogućih različitih vrijednosti za x_i , maksimalni period ili duljina ciklusa linearnog kongruentnog generatora iznosi m . Također, kako $x_{i-1} = 0$ nije dopušteno u multiplikativnom generatoru, maksimalni period u tom slučaju iznosi $m - 1$.

Period multiplikativnog kongruentnog generatora s multiplikatorom a i modulom m ovisi o najmanjoj pozitivnoj vrijednosti k koja zadovoljava

$$a^k \equiv 1 \pmod{m}. \quad (2.7)$$

Razlog tome je činjenica da se, kada je relacija zadovoljena, niz počinje ponavljati. Stoga, period ne može biti veći od k . Zbog Euler-Fermatova teorema³ vidimo kako period ne može biti veći od $\varphi(m)$. Za danu vrijednost m , tražimo a takav da je k iz (2.7) jednak $\varphi(m)$. Takav a zovemo *primitivan korijen modulo m* . Ako je m prost, broj primitivnih korijena modulo m je $\varphi(m - 1)$.

Na primjer, uzmimo $m = 31$ i $a = 7$, to jest,

$$x_i \equiv 7x_{i-1} \pmod{31},$$

i počnimo s $x_0 = 19$. Sljedeći brojevi u nizu su

$$9, 1, 7, 18, 2, 14, 5, 4, 28, 10, 8, 25, 20, 16, 19,$$

nakon čega se niz počinje ponavljati. Pri tome je period 15, a također imamo

$$7^{15} \equiv 1 \pmod{31},$$

to jest, 7 nije primitivan korijen modulo 31.

Ako uzmemo $x_0 = 19$, uz $m = 31$ i $a = 3$, proći ćemo 30 brojeva prije nego što dođemo do 19. Uzrok tome je taj što je 3 primitivan korijen modulo 31. Postoji $\varphi(30) = 8$ primitivnih korijena modulo 31.

Da bi generator slučajnih brojeva bio koristan u praktičnim situacijama, period mora biti reda veličine 10^9 , što znači da modul u LCG mora biti barem toliko velik. Vrijednosti modula u općoj uporabi kreću se između 10^9 i 10^{15} . Unatoč tome, period takvih generatora je relativno kratak u pogledu vrlo velikog broja simulacijskih eksperimenata te zbog brzine kojom računala kruže kroz puni period.

Postoje brojne varijacije osnovnog algoritma

$$x_i = f(x_{i-1}, \dots, x_{i-k}),$$

za stvaranje pseudoslučajnih brojeva. U jednostavnom linearnom kongruentnom generatoru (2.5) f je jednostavna linearna funkcija (to jest, $k = 1$) kombinirana s modularnim dijeljenjem. Neke varijante linearnog kongruenta koriste vrijednost $k > 1$:

³Ako su a i m relativno prosti pozitivni cijeli brojevi, tada $a^{\varphi(m)} \equiv 1 \pmod{m}$, gdje je $\varphi(m)$ Eulerova funkcija - broj brojeva u nizu $1, 2, \dots, m$ koji su relativno prosti s m .

$$x_i \equiv (a^T v_{i-1} + c) \pmod{m}, \quad 0 \leq x_i < m,$$

gdje je a k -dimenzionalan vektor konstanti, a v_{i-1} je k -dimenzionalan vektor $(x_{i-1}, \dots, x_{i-k})$. Takav generator za početnu vrijednost prima k -dimenzionalan vektor. Dva najkorištenija generatora prethodnog oblika su *višestruko rekurzivni* te *pomaknuti Fibbonacijev* generator.

Višestruko rekurzivni generatori:

Jednostavni dodatak multiplikativnom kongruentnom generatoru je korištenje prethodnih k vrijednosti za generiranje sljedeće:

$$x_i \equiv (a_1 x_{i-1} + a_2 x_{i-2} + \dots + a_k x_{i-k}) \pmod{m}. \quad (2.8)$$

Broj prethodno korištenih vrijednosti, k , naziva se *red* generatora. (Kada je $k = 1$, riječ je o multiplikativnom kongruentnom generatoru.)

Period danog generatora može biti mnogo veći nego kod jednostavnih multiplikativnih generatora. Za prosti m , Knuth (1998.) je pokazao da je, uz određene uvjete, maksimalni period $m^k - 1$. [1, str. 32]

Pomaknuti Fibbonacijev generator:

Za jednostavan Fibbonacijev niz vrijedi $x_{i+2} = x_{i+1} + x_i$. Generalizirajmo prethodnu rekurziju tako da, umjesto kombinacije uzastopnih članova, kombiniramo članove na većoj udaljenosti, pa stoga dobivamo

$$x_i \equiv (x_{i-j} + x_{i-k}) \pmod{m}. \quad (2.9)$$

Generator će raditi dobro u slučaju da su j , k i m odabrani korektno, u smislu da osiguravaju što veći period. Ako je m prost a $k > j$, tada period može biti najviše $m^k - 1$. Ako je modul potencija broja 2, na primjer 2^p , maksimalni period iznosi $(2^k - 1)2^{p-1}$.

Drugi način generalizacije Fibbonacijeve rekurzije je uporaba općenitijih binarnih operacija umjesto modula m . U općenitom pomaknutom Fibbonacijevom generatoru, započinjemo s x_1, x_2, \dots, x_k i stavljamo

$$x_i = (x_{i-j} \circ x_{i-k}),$$

gdje je \circ neki binarni operator, uz $0 \leq x_i \leq m - 1$ i $0 < j < k < i$.

Očita je generalizacija skalarnog linearnog kongruentnog generatora u ge-

erator pseudoslučajnih vektora:

$$x_i \equiv (Ax_{i-1} + c) \pmod{m}, \quad (2.10)$$

gdje su x_i, x_{i-1} i c vektori duljine d , a A je $d \times d$ matrica. Elementi vektora i matrice su cijeli brojevi između 1 i $m - 1$. Elementi vektora se tada skaliraju u interval $(0, 1)$ radi simulacije $\mathcal{U}(0, 1)$. Takav generator nazivamo *matrični kongruentni generator*. Slično kao i kod skalarnih generatora, često se uzima $c = 0$. Razlog korištenja matričnih generatora je stvaranje paralelnih nizova pseudoslučajnih brojeva ili uvođenje korelacijske strukture u slučajne vektore.

2.3 Nelinearni kongruentni generatori

Ako je funkcija f u osnovnoj kongruentnoj rekurziji

$$x_i = f(x_{i-1}, \dots, x_{i-k}),$$

linearna, ne samo da je izračun jednostavniji i obično brži, već se lakše provodi i analiza outputa. Ipak, promotrimo neke primjere nelinearnih generatora.

Inverzivni kongruentni generatori

Riječ je o generatorima koje su predstavili Eichenauer i Lehn (1986.), a koriste modularni multiplikativni inverz, ukoliko postoji, za generiranje sljedeće slučajne vrijednosti u nizu. Oblik generatora je

$$x_i \equiv (ax_{i-1}^- + c) \pmod{m}, \quad 0 \leq x_i < m, \quad (2.11)$$

gdje x^- predstavlja multiplikativni inverz od x modulo m . Multiplikativni inverz x^- od x modulo m definiran je za sve x različite od 0 koji su relativno prosti s m i za koje vrijedi

$$1 \equiv x_{i-1}^- x \pmod{m}.$$

Knuth (1998.) je predložio jednostavno poopćenje generatora (2.5):

$$x_i \equiv (dx_{i-1}^2 + ax_{i-1} + c) \pmod{m}, \quad 0 \leq x_i < m. \quad (2.12)$$

Blum, Blum i Shub (1996.) dali su varijantu generatora oblika

$$x_i \equiv x_{i-1}^2 \pmod{m}. \quad (2.13)$$

Razlika njihove metode ne očituje se samo u kompliciranijem načinu transformiranja nizova cijelih brojeva. Oni formiraju niz outputa kao dijelove $b_1b_2b_3\cdots$, gdje je $b_i = 0$ ako je x_i paran, a $b_i = 1$ inače. Pokazali su da rezultat ovog generatora nije predvidiv u polinomijalnom vremenu bez poznavanja p_1 i p_2 , pri čemu je $m = p_1p_2$, p_1 i p_2 različiti prosti brojevi, svaki kongruentan 3 modulo 4. Upravo zbog te nepredvidivosti, ovaj generator ima mogućnost značajne uporabe u kriptografiji.

3 Generiranje slučajnih varijabli

S upoznatim generatorima slučajnih brojeva ubuduće pretpostavljamo dostupnost idealnog niza slučajnih brojeva. Preciznije, pretpostavljamo da je moguće simulirati idealan niz slučajnih brojeva iz $(0, 1)$ od kojih je svaki realizacija uniformne slučajne varijable s funkcijom distribucije

$$P(U_i \leq u) = \begin{cases} 0, & u < 0 \\ u, & 0 \leq u \leq 1 \\ 1, & u > 1, \end{cases}$$

to jest, od kojih je svaka varijabla uniformno distribuirana na $(0, 1)$. Mnogo simulacija zahtijeva rad s uzorcima slučajnih varijabli koje imaju distribuciju različitu od uniformne. Tipična simulacija koristi metode za transformaciju uzoraka iz uniformne distribucije u uzorke iz drugih distribucija. U ovom ćemo se poglavlju baviti s dvije najkorištenije tehnike: *metodom inverzne transformacije* i *metodom prihvatanja-odbijanja*.

No, prije toga, definirat ćemo neke pojmove koji će nam biti korisni u kasnijem radu, poput slučajnog vektora, jednostavnog slučajnog uzorka, statističkog modela i drugih, kako bi objasnili da su podaci koje simuliramo iz distribucije zapravo jedna realizacija takvog slučajnog uzorka.

Definicija 3.1. *Neka je dan vjerojatnosni prostor (Ω, \mathcal{F}, P) . Funkciju (X_1, \dots, X_n) koja svakom ishodu pokusa pridružuje uređenu n -torku realnih brojeva (x_1, \dots, x_n) zovemo n -dimenzionalan slučajni vektor ako vrijedi:*

$$\{X_1 \leq x_1\} \cap \cdots \cap \{X_n \leq x_n\} \in \mathcal{F}$$

za svaki $x_1 \in \mathbb{R}, \dots, x_n \in \mathbb{R}$.

Definicija 3.2. *Neka je (Ω, \mathcal{F}, P) vjerojatnosni prostor i (X_1, \dots, X_n) slučajni vektor. Funkciju*

$$F : \mathbb{R} \rightarrow [0, 1],$$

$$F(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n)$$

zovemo funkcija distribucije slučajnog vektora (X_1, \dots, X_n) .

Neka je (X, Y) neprekidan slučajni vektor s funkcijom gustoće f i funkcijom distribucije F . Neka su f_X i f_Y njegove marginalne gustoće, a F_X i F_Y funkcije distribucije. Reći ćemo da su slučajne varijable X i Y nezavisne ako za svaki $(x, y) \in \mathbb{R}^2$ vrijedi $F(x, y) = F_X(x)F_Y(y)$. Tada je i $f(x, y) = f_X(x)f_Y(y)$ za svaki $(x, y) \in \mathbb{R}^2$. Ako prethodna jednakost za gustoće vrijedi za sve $(x, y) \in \mathbb{R}^2$ osim eventualno na skupu vjerojatnosti nula, tada su X i Y nezavisne slučajne varijable.

Zadati statistički model znači opisati poznate karakteristike slučajnog vektora za kojega smatramo da naši podaci čine jednu realizaciju. Kako je slučajni vektor određen svojom funkcijom distribucije, statističkim je modelom opisano ono što je unaprijed poznato o funkciji distribucije slučajnog vektora kojim modeliramo podatke, odnosno statistički model jest familija funkcija distribucije koja se uzima u obzir za zaključivanje u danom problemu.

Definicija 3.3. *Statistički model zovemo model jednostavnoga slučajnog uzorka iz funkcije distribucije F ako za slučajni vektor (X_1, \dots, X_n) , čiju realizaciju čine podaci (x_1, \dots, x_n) , vrijedi:*

- *slučajne varijable X_1, \dots, X_n su nezavisne,*
- *sve slučajne varijable X_1, \dots, X_n imaju istu funkciju distribucije F .*

Kod takvih modela promatranu veličinu smatramo slučajnom varijablom s funkcijom distribucije F , a vrijednosti varijable izmjerene na jedinkama iz uzorka nezavisne su realizacije te slučajne varijable. U nastavku rada ćemo koristiti termin slučajni uzorak za slučajni vektor (X_1, \dots, X_n) , a termin uzorak za njegovu realizaciju (x_1, \dots, x_n) , tj. podatke.

3.1 Metoda inverzne transformacije

Teorem 3.1. *Neka je dana neprekidna funkcija distribucije F na \mathbb{R} s inverznom funkcijom F^{-1} definiranom kao*

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}, \quad 0 < u < 1.$$

Ako je U uniformna slučajna varijabla na $(0, 1)$, tada F^{-1} ima distribuciju F . Dodatno, ako je X slučajna varijabla s funkcijom distribucije F , tada je $F(X)$ uniformno distribuirana na $(0, 1)$.

Dokaz. Prva tvrdnja teorema slijedi iz činjenice da je $\forall x \in \mathbb{R}$

$$P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x).$$

Prva jednakost rezultat je primjene monotone funkcije F na obje strane, dok druga jednakost proizlazi iz $P(U \leq y) = y$, budući da je $U \sim \mathcal{U}(0, 1)$.

Druga tvrdnja teorema slijedi iz rezultat da je za sve $0 < u < 1$,

$$P(F(X) \leq u) = P(X \leq F^{-1}(u)) = F(F^{-1}(u)) = u.$$

□

Stoga, za generiranje slučajne varijable X s funkcijom distribucije F koristimo sljedeći algoritam.

Algoritam 3.1 (Metoda inverzne transformacije).

1. *Generiraj U iz $\mathcal{U}(0, 1)$*
2. *Vrati $X = F^{-1}(U)$.*

Ulazni podatak U možemo interpretirati kao slučajni percentil. Ako je F neprekidna i $X \sim F$, tada je jednako vjerojatno da će se X realizirati između, na primjer, 10. i 20. percentila od F kao između 70. i 80. percentila. Drugim riječima, percentil u koji X (to jest, $F(X)$) pada je uniformno distribuiran.

Ilustrirajmo metodu pomoću sljedećih primjera.

Primjer 3.1.

Generirajmo slučajni uzorak iz slučajne varijable s funkcije gustoće

$$f(x) = \begin{cases} 2x, & 0 \leq x \leq 1 \\ 0, & \text{inače.} \end{cases}$$

Funkcija distribucije je

$$F(x) = \begin{cases} 0, & x < 0 \\ \int_0^x 2y dy, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases} = \begin{cases} 0, & x < 0 \\ x^2, & 0 \leq x \leq 1 \\ 1, & x > 1. \end{cases}$$

Tada imamo

$$X = F^{-1}(U) = \sqrt{U}.$$

Prema tome, za generiranje slučajne varijable X s gornjom funkcijom gustoće, prvo generiramo uzorak iz $\mathcal{U}(0, 1)$ a zatim uzmemo kvadratni korijen svakog elementa uzorka i tako dobijemo uzorak iz slučajne varijable X .

Primjer 3.2. Eksponencijalna distribucija

Eksponencijalna distribucija s parametrom $\frac{1}{\theta}$ zadana je funkcijom distribucije

$$F(x) = \begin{cases} 1 - e^{-\frac{x}{\theta}}, & x \geq 0 \\ 0, & \text{inače.} \end{cases}$$

Invertiranje vraća $X = -\theta \log(1 - U)$, što se može implementirati i kao $X = -\theta \log(U)$, budući da U i $1 - U$ imaju jednaku distribuciju.

Dokažimo tvrdnju iz prethodnog primjera koja kaže da slučajne varijable U i $1 - U$ imaju jednaku distribuciju. Neka je $U \sim \mathcal{U}(0, 1)$. Tada, za $0 \leq u \leq 1$ imamo $F_U(u) = u$. Označimo s $V = 1 - U$. Tražimo $F_V(v)$ za $0 \leq v \leq 1$. Imamo

$$F_V(v) = P(1 - U \leq v) = P(U \geq 1 - v) = 1 - P(U \leq 1 - v) = 1 - (1 - v) = v.$$

Još treba pokazati da je $F_V(v) = 0$ za $v < 0$ i $F_V(v) = 1$ za $v > 1$. Za $v < 0$ je

$$P(V \leq v) = P(1 - U \leq v) = P(U \geq 1 - v).$$

Kako je $1 - v > 1$, slijedi da je $F_V(v) = 0$. Analogno, $F_V(v) = 1$ za $v > 1$.

Algoritam 3.2 (Metoda inverzne transformacije za diskretnu distribuciju).

1. Generiraj $U \sim \mathcal{U}(0, 1)$
2. Nađi najmanji pozitivni cijeli broj k takav da je $U \leq F(x_k)$ i vrati $X = x_k$.

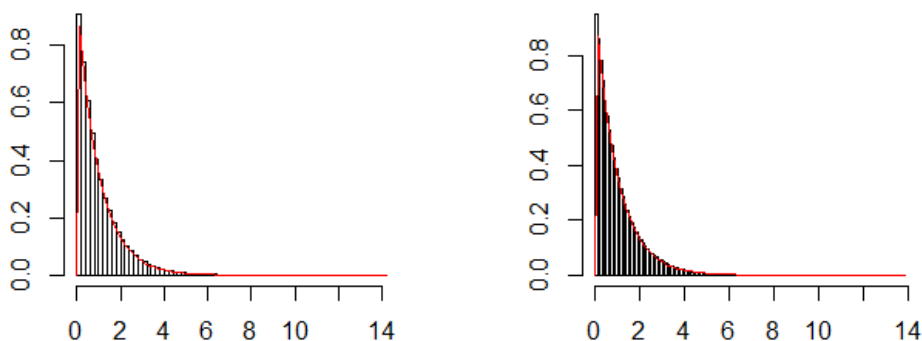
Primjer 3.3.

Uzmimo za primjer da je $X \sim \text{Exp}(1)$. Analogno prethodnom primjeru,

$$F(x) = \begin{cases} 1 - e^{-x}, & x > 0 \\ 0, & \text{inače.} \end{cases}$$

Rješavajući jednadžbu $u = 1 - e^{-x}$ po x , imamo $x = -\log(1 - u)$. Stoga, ako je $U \sim \mathcal{U}(0, 1)$, tada je $X = -\log(U) \sim \text{Exp}(1)$.

Napravili smo R kod koji uspoređuje izlazne podatke dobivene metodom inverzne transformacije s izlaznim podacima dobivenih korištenjem ugrađene funkcije `rexp`, koja generira podatke iz eksponencijalne distribucije.



Slika 1: Histogram podataka iz eksponencijalne slučajne varijable u ovisnosti o metodi generiranja

Lijevi histogram dobiven je metodom inverzne transformacije, a desni koristeći R naredbu `rexp`. Samim promatranjem oba histograma teško možemo uočiti razliku u metodi generiranja, što ćemo pokušati potvrditi provođenjem *two-sample Kolmogorov-Smirnovljevog testa*, odnosno testirat ćemo nultu hipotezu o jednakosti distribucije dva uzorka. U našem slučaju, radi se o uzorcima dobivenim metodom inverzne transformacije i uzorcima dobivenim korištenjem R funkcije. Dobivena p -vrijednost 0.5853, sugerira kako nemamo razloga odbaciti nultu hipotezu o jednakosti distribucije dva generirana uzorka.

Općenito, metoda inverzne transformacije zahtijeva da temeljna funkcija distribucije, F , postoji u obliku za koji odgovarajuća inverzna funkcija F^{-1} može biti pronađena analitički ili algoritamski. Drugim riječima, dovoljno je da se može pronaći inverz za restrikciju funkcije distribucije na području gdje ona prima vrijednosti veće od nule. Primjenjive distribucije su, na primjer, eksponencijalna, uniformna, Weibullova i Cauchyjeva. Za brojne druge distribucije je teško ili pak nemoguće naći inverznu transformaciju, to jest, riješiti

$$F(x) = \int_{-\infty}^x f(t)dt = u$$

obzirom na x . Čak i u slučaju kada F^{-1} postoji u eksplicitnom obliku, ova metoda ne mora nužno biti najefikasnija metoda za generiranje slučajnih varijabli.

3.2 Metoda prihvatanja-odbijanja

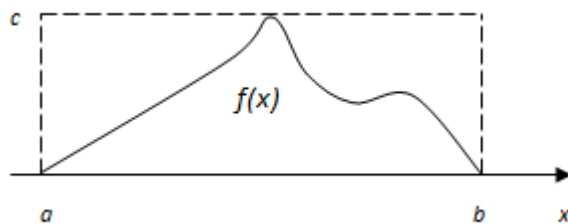
Riječ je o metodi koja pripada među najprimjenjivije mehanizme za generiranje slučajnih uzoraka. Prethodnu metodu mogli bi svrstati u direktne metode, u smislu da se bave direktno s funkcijom distribucije slučajne varijable koju želimo generirati. S druge strane, metodu prihvatanja-odbijanja smatramo indirektnom metodom. Osmislio ju je John von Neumann, a funkcionira na način da uzorak iz ciljane distribucije generira tako što prvo generira kandidate iz prikladnije distribucije, a zatim odbacuje slučajni podskup generiranih kandidata. Pojam prikladnije distribucije u ovom se kontekstu odnosi na distribucije iz kojih znamo i možemo generirati uzorak. Mehanizam odbacivanja dizajniran je kako bi prihvaćeni uzorci uistinu bili distribuirani na željeni način.

Za početak pretpostavimo da je ciljana funkcija gustoće f (funkcija gustoće iz koje želimo dobiti uzorak) ograničena na nekom segmentu $[a, b]$, a 0 izvan tog segmenta (Slika 1). Označimo s

$$c = \sup\{f(x) : x \in [a, b]\}.$$

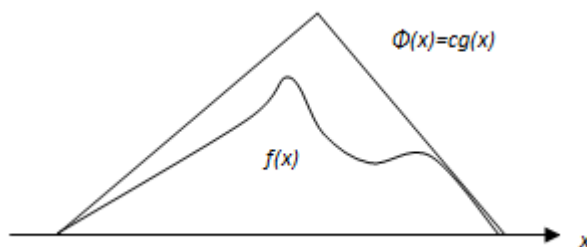
U ovom slučaju, generiranje slučajne varijable $Z \sim f$ je jasno, a rezultat je sljedećih koraka:

1. Generiraj $X \sim \mathcal{U}(a, b)$
2. Generiraj $Y \sim \mathcal{U}(0, c)$ nezavisno od X
3. Ako je $Y \leq f(X)$, vrati $Z = X$. Inače, povratak na korak 1.



Slika 2: Metoda prihvaćanja-odbijanja

Valja napomenuti kako je svaki generirani vektor (X, Y) uniformno distribuiran na pravokutniku $[a, b] \times [0, c]$. Stoga, prihvaćeni par (X, Y) je uniformno distribuiran ispod grafa f . To implicira da distribucija prihvaćenih vrijednosti od X ima željenu funkciju gustoće f .

Slika 3: Metoda prihvaćanja-odbijanja s funkcijom ograničavanja $\phi(x)$

Generalizaciju prethodnog razmatranja možemo izvršiti na sljedeći način: neka je g proizvoljna funkcija takva da $\phi(x) = cg(x)$ odozgo ograničava funkciju $f(x)$ za neku konstantu c (Slika 2), to jest, da je $\phi(x) \geq f(x)$ za sve x . Nužan uvjet je $c \geq 1$. Funkciju g zovemo *predložena* funkcija gustoće i pretpostavljamo da je iz nje jednostavno generirati slučajni uzorak.

Algoritam 3.3 (Metoda prihvatanja-odbijanja).

1. Generiraj X iz $g(x)$
2. Generiraj $Y \sim \mathcal{U}(0, cg(X))$.
3. Ako je $Y \leq f(X)$, vrati $Z = X$. Inače, povratak na korak 1.

Teorijska podloga ove metode dana je sljedećim teoremom.

Teorem 3.2. *Slučajna varijabla generirana algoritmom 3.3 ima željenu funkciju gustoće $f(x)$.*

Dokaz. Definirajmo sljedeće podskupove:

$$\mathcal{A} = \{(x, y) : 0 \leq y \leq cg(x)\} \quad \text{i} \quad \mathcal{B} = \{(x, y) : 0 \leq y \leq f(x)\},$$

koji predstavljaju područja ispod krivulja $cg(x)$ i $f(x)$, redom. Koraci 1 i 2 algoritma 3.3 ukazuju da je slučajni vektor (X, Y) uniformno distribuiran na \mathcal{A} . Kako bi to provjerili, označimo s $q(x, y)$ funkciju gustoće od (X, Y) , a s $q(y|x)$ uvjetnu funkciju gustoće od Y uz $X = x$. Tada imamo

$$q(x, y) = \begin{cases} g(x)q(y|x), & (x, y) \in \mathcal{A} \\ 0, & \text{inače.} \end{cases}$$

Iz koraka 2 vidimo da je $q(x, y) = 1/cg(x)$, za $y \in [0, cg(x)]$, a 0 inače. Stoga, $q(x, y) = c^{-1}$ za svaki $(x, y) \in \mathcal{A}$.

Neka (X^*, Y^*) označava prvu prihvaćenu vrijednost, to jest, prvu koja se nalazi u \mathcal{B} . Budući je vektor (X, Y) uniformno distribuiran na \mathcal{A} , tada je vektor (X^*, Y^*) uniformno distribuiran na \mathcal{B} . Također, funkcija gustoće od (X^*, Y^*) na \mathcal{B} je jednaka 1. Stoga, marginalna funkcija gustoće od $Z^* = X$ je

$$\int_0^{f(x)} 1 dy = f(x).$$

□

Vjerojatnost prihvatanja prilikom svakog generiranja iznosi $1/c$. Budući da su pokušaji međusobno nezavisini, broj kandidata generiranih do prvog prihvatanja je geometrijski distribuiran s očekivanjem c . Poželjno je, stoga, imati vrijednost c što bližu 1 (što će se dogoditi kada je $g(x)$ blizu $f(x)$). Čvršća ograničenja na željenu gustoću f rezultirat će manjim brojem krivih

uzoraka iz g .

Kažemo kako efikasnost algoritma 3.3 iznosi $1/c$. Naime,

$$P((X, Y) \text{ prihvaćen}) = \frac{\text{područje } \mathcal{B}}{\text{područje } \mathcal{A}} = \frac{1}{c}.$$

Često se koristi blago modificirana verzija prethodnog algoritma. Naime, uzevši u obzir da je $Y \sim \mathcal{U}(0, cg(X))$ u koraku 2 isto što i $Y = Ucg(X)$, gdje je $U \sim \mathcal{U}(0, 1)$, $Y \leq f(X)$ u koraku 3 možemo pisati kao $U \leq f(X)/cg(X)$. Tako dobivamo izmijenjenu verziju algoritma 3.3:

Algoritam 3.4 (Modificirana metoda prihvaćanja-odbijanja).

1. Generiraj X iz $g(x)$
2. Generiraj $U \sim \mathcal{U}(0, 1)$ nezavisno od X
3. Ako je $U \leq f(X)/cg(X)$, vrati $Z = X$. Inače, povratak na korak 1.

Drugim riječima, generiramo X iz $g(x)$ i prihvaćamo ju s vjerojatnošću $f(X)/cg(X)$; u suprotnom, odbacujemo X i pokušavamo ponovno. Prednosti modificirane verzije su generiranje uniformne slučajne varijable nezavisno od X i poznavanje vjerojatnosti prihvaćanja.

Primjer 3.4.

Pretpostavimo da želimo simulirati uzorak iz slučajne varijable iz $\mathcal{Be}(\alpha, \beta)$ metodom prihvaćanja-odbijanja.

Funkcija gustoće beta distribucije, za $0 \leq x \leq 1$ i parametre $\alpha, \beta > 0$ definirana je izrazom

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1},$$

gdje je B beta funkcija, još poznata pod nazivom Eulerov integral prve vrste, definirana s

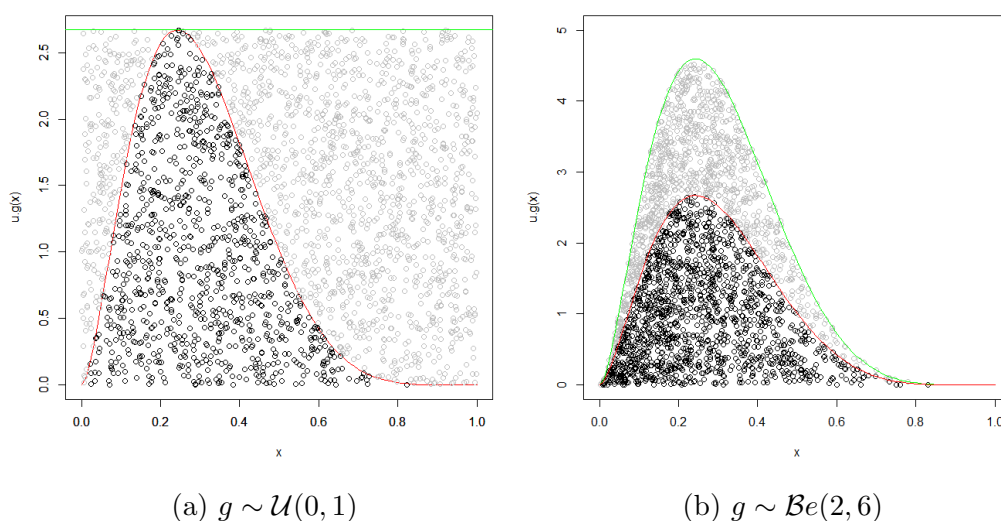
$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt,$$

za $\text{Re}(x), \text{Re}(y) > 0$. U prvom slučaju za preloženu distribuciju uzet ćemo $\mathcal{U}(0, 1)$, uz $\alpha, \beta > 1$. Gornja granica iz algoritma, c , tada je maksimum beta distribucije na intervalu $(0, 1)$. Tu vrijednost dobili smo koristeći ugrađenu R funkciju optimize. Kako je funkcija gustoće g kandidata jednaka 1 na $(0, 1)$,

predložena vrijednost X je prihvaćena ako $U \cdot c < f(X)$, to jest, ako se $U \cdot c$ realizira unutar beta gustoće f . Generiranje $U \sim \mathcal{U}(0, 1)$ uz množenje s c ekvivalentno je generiranju $U \sim (0, c)$.

Za $\alpha = 2.7$ i $\beta = 6.3$, lijevi dio slike 4 prikazuje rezultat generiranja 2500 parova (X, U) iz $\mathcal{U}(0, 1) \times \mathcal{U}(0, c)$. Crne točke $(X, Ug(X))$ koje se nalaze ispod gustoće f su one vrijednosti za koje prihvaćamo $Z = X$, a odbacujemo vrijednosti obojane sivo koje se nalaze izvan. Kako je vjerojatnost prihvaćanja simulacije dana s $1/c$, u našem slučaju, za $c = 2.67$, prihvaćamo oko 37% vrijednosti.

U drugom slučaju, reprezentiranom desnim dijelom slike 4, za predloženu distribuciju smo uzeli $\mathcal{Be}(2, 6)$ i dobili oko 58% prihvaćenih simuliranih vrijednosti. U oba slučaja, prikazane su funkcije f (crveno) i $c \cdot g$ (zeleno).



Slika 4: Generiranje slučajnih varijabli $X \sim \mathcal{Be}(2.7, 6.3)$

4 Monte Carlo metode

Monte Carlo simulacija je postupak eksperimentiranja sa slučajnim brojevima kako bi se procijenili matematički izrazi, poput konačnih integrala, rješenja sustava jednadžbi ili kompliciranijih matematičkih modela. U većini analiza matematičkih izraza mogu se preferirati standardne procjene iz numeričke analize, ali Monte Carlo metode nude alternativni pristup koji je

ponekad jedini prikladan za lako korištenje i kontroliranje. Na primjer, čest slučaj je korištenje Monte Carlo metode za procjenu integrala na domenama viših dimenzija.

Ponekad je u modelu prisutna slučajnost, pa cilj Monte Carlo proučavanja postaje otkrivanje distribucije neke varijable. Zamislimo da nas zanima distribucija "mid-rangea" (aritmetička sredine minimalne i maksimalne vrijednosti) gama distribucije. Ulazni podatak je varijabla s gama distribucijom, a izlazni je mid-range statistika. Umjesto procjenjivanja parametara mid-rangea, poput očekivanja i varijance, Monte Carlo kao cilj može imati procjenu funkcije gustoće mid-rangea. Tada je subjekt procjene neprekidna funkcija.

Brojna su praktična pitanja koja se moraju uzeti u obzir prilikom provođenja Monte Carlo metode. Analiza mora biti obavljena kao skup odvojenih simulacija. U tom slučaju, nužno je da simulacije ne koriste isprepletene nizove slučajnih brojeva, u smislu da se zahtjeva rad s kompletno različitim nizovima. Rezultat bilo kojeg znanstvenog eksperimenta mora se moći reproducirati, a metoda mora biti efikasna. Monte Carlo metoda započinje s identifikacijom slučajne varijable na način da je očekivana vrijednost neke funkcije te slučajne varijable parametar u problemu kojeg rješavamo. Dobar primjer Monte Carlo metode prikazat ćemo u sljedećem potpoglavlju.

4.1 Procjena integrala

U svom najjednostavnijem obliku, Monte Carlo simulacija je procjena konačnog integrala

$$\theta = \int_D f(x)dx$$

identificiranjem slučajne varijable Y s vrijednostima u D , funkcije gustoće $p(y)$ i funkcije g takve da je matematičko očekivanje od $g(Y)$ upravo θ :

$$\begin{aligned} E(g(Y)) &= \int_D g(y)p(y)dy \\ &= \int_D f(y)dy \\ &= \theta. \end{aligned}$$

Promotrimo slučaj u kojem je D segment $[a, b]$, $Y \sim \mathcal{U}[a, b]$, a $g = f$. U tom slučaju,

$$\theta = (b - a)\mathbb{E}(f(Y)).$$

Problem procjene integrala postaje uobičajen statistički problem procjene očekivanja, $\mathbb{E}(f(Y))$.

Definicija 4.1. *Neka je $\mathcal{P} = \{F_\theta : \theta \in \Theta \subseteq \mathbb{R}^k\}$ parametarski statistički model, gdje je θ k -dimenzionalan parametar, a $\Theta \subseteq \mathbb{R}^k$ prostor parametara, tj. skup svih dozvoljenih vrijednosti nepoznatog parametra θ . Neka je (X_1, \dots, X_n) slučajni vektor s distribucijom iz \mathcal{P} i $t : \mathbb{R}^n \rightarrow \Theta$. Slučajni vektor $T = t(X_1, \dots, X_n)$ jest procjenitelj za θ .*

Općenito, ako razmatramo problem procjene očekivanja jednostavnog slučajnog uzorka (X_1, \dots, X_n) , možemo iskoristiti aritmetičku sredinu uzorka

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

U našem slučaju prirodno uzimamo slučajni uzorak i koristimo uzoračko očekivanje. Za uzorak veličine m , procjenitelj θ je

$$\hat{\theta} = (b - a) \frac{\sum_{i=1}^m f(y_i)}{m}, \quad (4.1)$$

gdje su y_i vrijednosti slučajnog uzorka iz uniformne distribucije na (a, b) . Procjenitelj je nepristran:

$$\begin{aligned} \mathbb{E}(\hat{\theta}) &= (b - a) \frac{\sum_{i=1}^m \mathbb{E}(f(Y_i))}{m} \\ &= (b - a)\mathbb{E}(f(Y)) \\ &= \int_a^b f(x) dx. \end{aligned}$$

Varijanca je

$$\begin{aligned} \text{Var}(\hat{\theta}) &= (b - a)^2 \frac{\sum_{i=1}^m \text{Var}(f(Y_i))}{m^2} \\ &= \frac{(b - a)^2}{m^2} \text{Var}(f(Y)) \\ &= \frac{(b - a)}{m} \int_a^b \left(f(x) - \int_a^b f(t) dt \right)^2 dx. \end{aligned} \quad (4.2)$$

Integral u jednadžbi (4.2) je mjera glatkoće⁴ funkcije. U ovom slučaju, glatkoću mjerimo kao L_2 normu razlike funkcije od svoje integrirane vrijednosti. Opisanu metodu ponekad nazivamo *grubi Monte Carlo*.

Pretpostavimo da početni integral može biti zapisan kao

$$\begin{aligned}\theta &= \int_D f(x)dx \\ &= \int_D g(x)p(x)dx,\end{aligned}$$

gdje je $p(x)$ funkcija gustoće na D , što može zahtijevati određene prilagodbe. Naime, uz ranije promatrani uniformni uzorak, $D = (a, b)$, a i b moraju biti konačni te $p(x) = 1/(b - a)$. Pretpostavimo još da možemo generirati m slučajnih vrijednosti y_i iz distribucije s gustoćom p . Tada je naš procjenitelj

$$\hat{\theta} = \frac{\sum_{i=1}^m g(y_i)}{m}. \quad (4.3)$$

Primjetimo da se procjenitelj (4.3) odnosi samo na integrale na općenitim domenama, dok (4.1) zahtjeva konačne intervale. Također, bitna razlika je ta što će varijanca procjenitelja (4.3) po svoj prilici biti manja od varijance u (4.1), tj. procjenitelj će biti efikasniji. Naime, u ovom slučaju imamo

$$\begin{aligned}\text{Var}(\hat{\theta}) &= \frac{\sum_{i=1}^m \text{Var}(g(Y_i))}{m^2} \\ &= \frac{1}{m^2} \text{Var}(g(Y)).\end{aligned}$$

4.2 Eksperimentalna greška u Monte Carlo metodama

Monte Carlo metode baziraju se na prikupljanju uzoraka, stoga procjene nastale iz tih metoda imaju prateće uzoračke greške. Činjenica da procjena nije jednaka očekivanoj vrijednosti (pretpostavljajući da je procjenitelj nepristran) nije pogreška ili propust, već je rezultat varijance slučajnih podataka. Varijabilnost slučajnih podataka dovodi do eksperimentalne greške, kao i kod svih drugih znanstvenih eksperimenata u kojima je slučajnost prepoznata komponenta.

Jednako kao u bilo kojem statističkom problemu procjenjivanja, procjenu

⁴Funkcija je glatka ako ima neprekidnu derivaciju.

treba pratiti procjena varijance. Procjena varijance procjenitelja obično je samo uzoračka varijanca izračunatih vrijednosti procjenitelja. Kako bi ustanovili približni pouzdani interval za integral kojeg procjenjujemo, možemo koristiti kvadratni korijen varijance (to jest, standardnu devijaciju) Monte Carlo procjenitelja. U tom slučaju, granice intervala bi sadržavale nepoznate termine varijance. Osim toga, mogli bismo procijeniti varijancu procjenitelja koristeći isti uzorak koji smo koristili za procjenu integrala. Važno svojstvo standardne devijacije Monte Carlo procjenitelja konačnog integrala je nezavisnost broja procjena funkcije od dimenzionalnosti integrala.

Glavni razlog analize varijance procjenitelja u Monte Carlu je odlučivanje treba li povećati veličinu uzorka. Često se ta veličina određuje na način da duljina pouzdanog intervala procjenitelja odgovara zahtjevu za maksimalnom duljinom. Eksperimentalna greška u Monte Carlu treba biti tretirana s jednakom pozornosti kao i greške u drugim eksperimentima, budući da se prenosi kroz buduće izračune. Također, greška određuje granicu broja značajnih znamenki u numeričkim rezultatima.

4.3 Varijanca Monte Carlo procjenitelja

Varijanca Monte Carlo procjenitelja ima važnu ulogu u ocjenjivanju kvalitete procjene integrala. Izraz za varijancu, kao u (4.2), može biti prilično kompliciran. Stoga, trebamo metode za procjenu varijance Monte Carlo procjenitelja.

Monte Carlo procjenitelj obično ima oblik procjenitelja od θ u (4.1):

$$\hat{\theta} = c \frac{\sum_{i=1}^m f_i}{m}.$$

Varijanca procjenitelja ima oblik jednadžbe (4.2):

$$\text{Var}(\hat{\theta}) = c \int \left(f(x) - \int f(t) dt \right)^2 dx.$$

Procjenitelj varijance, u oznaci $\hat{\sigma}^2$, je

$$\hat{\sigma}^2(\hat{\theta}) = c^2 \frac{\sum_{i=1}^m (b_i - \bar{b})^2}{m - 1}, \quad (4.4)$$

gdje je $\bar{b} = \frac{1}{m} \sum_{i=1}^m b_i$. Procjenitelj je prikladan samo ako su elementi skupa slučajnih varijabli $\{B_i\}$, za koje imamo opažanja $\{b_i\}$, nezavisni i time imaju

korelaciju 0.

Ako $\{B_i\}$ nemaju korelaciju 0, procjenitelj (4.4) ima očekivanu vrijednost koja sadrži korelacije, to jest, nije nepristran za procjenu $\text{Var}(\hat{\theta})$, što je situacija koja se pojavljuje često u simulacijama. U mnogim procesima, opažanja su "nezavisnija" od opažanja koja su udaljenija od njih unutar niza, nego od opažanja koja su im bliže. Uobičajena metoda za procjenu varijance u nizu opažanja koja nisu nezavisna je korištenje prosjeka uzastopnih podnizova koji su dovoljno dugački da opažanja u jednom podnizu budu gotovo nezavisna od opažanja u drugom podnizu. Prosjek podnizova zove se *grupni prosjek*⁵. Ako je $B_1, \dots, B_b, B_{b+1}, \dots, B_{2b}, B_{2b+1}, \dots, B_{kb}$ niz slučajnih varijabli takav da je korelacija između B_i i B_{i+b} približno 0, procjena varijance prosjeka, \bar{B} , $m = kb$ slučajnih varijabli može biti izračunata na sljedeći način

$$\begin{aligned} \text{Var}(\bar{B}) &= \text{Var}\left(\frac{1}{m} \sum_{i=1}^m B_i\right) \\ &= \text{Var}\left(\frac{1}{k} \sum_{j=1}^k \left(\frac{1}{b} \sum_{i=(j-1)b+1}^{jb} B_i\right)\right) \\ &\approx \frac{1}{k^2} \sum_{j=1}^k \text{Var}\left(\frac{1}{b} \sum_{i=(j-1)b+1}^{jb} B_i\right) \\ &\approx \frac{1}{k} \text{Var}(\bar{B}_b), \end{aligned}$$

gdje je \bar{B}_b prosjek grupe duljine b . Ako su grupe dovoljno dugačke, razumno je pretpostaviti da prosjeci imaju zajedničku varijancu. Procjenitelj varijance od \bar{B}_b je standardna varijanca od k opažanja, $\bar{b}_1, \dots, \bar{b}_k$:

$$\frac{\sum_{i=1}^k (\bar{b}_i - \bar{b})^2}{k-1}.$$

Stoga, batch-means procjenitelj varijance od \bar{B} je

$$\hat{\sigma}^2(\bar{B}) = \frac{\sum_{i=1}^k (\bar{b}_i - \bar{b})^2}{k(k-1)}. \quad (4.5)$$

Ovaj procjenitelj treba koristiti prilikom rada s podacima koji nisu nezavisni, poput vremenskih nizova, ili simulacija koje koriste Markovljev lanac, o čemu

⁵eng. batch means[1]

će više govora biti u sljedećem poglavlju. Veličina dijelova uzorka treba biti što je moguće manja, a pri tome i dalje imati nezavisna očekivanja. Testovi nezavisnosti za \bar{B}_b mogu biti prikladni za odabir veličine grupe. Grupni prosjeci su korisni u procjeni varijance kad god se Markovljev lanac koristi u generiranju slučajnih vrijednosti.

Varijancu možemo procjenjivati iz rezultata različitih eksperimenata, budući da ti rezultati trebaju biti nekorelirani. Poželjno je izvršiti zasebne eksperimente kako bi se dobila što pouzdanija procjena varijance.

U poglavlju 2 smo predstavili neke različite generatore slučajnih brojeva i uporabu više od jedne početne vrijednosti. Promotrimo klasični linearan model za opažanja y_{ijk} u eksperimentu

$$y_{ijk} = \mu + g_i + s_{i(j)} + e_{ij(k)},$$

gdje je μ veličina o kojoj želimo donositi zaključke, g_i efekt na opažanje od i -tog generatora, $s_{i(j)}$ efekt j -te početne vrijednosti u sklopu i -tog generatora te $e_{ij(k)}$ slučajna greška na k -tom opažanju u okviru ij para generatora i početne vrijednosti. Ako su svi generatori "savršeni", tada je $g_i = s_{i(j)} = 0$, za sve i i j . Prosjek svih opažanja koristeći ij par generatora i početne vrijednosti, $\bar{y}_{ij\cdot}$, ne bi trebao prikazivati sustavne oscilacije. Uzoračka varijanca od $\bar{y}_{ij\cdot}$ treba biti u skladu s prosjekom odgovarajuće veličine iz distribucije s varijancom greške $e_{ij(k)}$.

Cilj uzimanja uzorka je smanjenje varijance procjenitelja a da se pri tome sačuvaju druge dobre osobine, poput nepristranosti. Smanjenje varijance daje statistički efikasnije procjenitelje. Uobičajen pristup smanjenju varijance je korištenje dodatnih informacija o problemu radi smanjenja efekta slučajnog uzimanja uzorka na varijancu opažanja. Posvetit ćemo se metodi analitičke redukcije.

Prvo načelo procjene je da se koriste sve poznate veličine kako bi se poboljšala procjena. Na primjer, pretpostavimo da imamo problem procjene integrala

$$\int_D f(x) dx$$

Monte Carlo metodama. Pretpostavimo da su D_1 i D_2 takvi da je $D_1 \cup D_2 = D$ i $D_1 \cap D_2 = \emptyset$, te promotrimo prikaz integrala

$$\begin{aligned}\theta &= \int_{D_1} f(x)dx + \int_{D_2} f(x)dx \\ &= \theta_1 + \theta_2.\end{aligned}$$

Pretpostavimo nadalje da je dio rastava originalnog problema poznat, to jest, pretpostavimo da znamo θ_1 . Vrlo je vjerojatno kako bi bilo bolje koristiti Monte Carlo metode samo za procjenu θ_2 , a zatim kao procjenu od θ smatrati sumu poznatog θ_1 i procijenjene vrijednosti θ_2 . Intuitivno se ovo čini očiglednim i obično je dobro ukoliko ne postoji neki odnos između $f(x_1)$ i $f(x_2)$ (na primjer, jedna raste kako druga pada), gdje je $x_1 \in D_1$, a $x_2 \in D_2$. Ako postoji neki odnos, procjenu $\hat{\theta}_2$ od θ_2 moguće je poboljšati koristeći transformaciju istih slučajnih brojeva korištenih za $\hat{\theta}_1$, procjenu od θ_1 .

Sada razmotrimo drukčiji zapis integrala, u kojem je f prikazana kao suma $g + h$, gdje g i h imaju isti predznak, odnosno obje su ili pozitivne ili negativne. Imamo

$$\begin{aligned}\theta &= \int_D (g(x) + h(x))dx \\ &= \int_D g(x)dx + \int_D h(x)dx \\ &= \theta_3 + \theta_4,\end{aligned}$$

te pretpostavimo da je dio rastava, recimo θ_3 , poznat. U ovom slučaju, korištenje poznate vrijednosti $\int_D g(x)dx$ pomoći će samo ako se $g(x)$ mijenja slično $f(x)$. Bolje je koristiti Monte Carlo metode samo za procjenu θ_4 a za procjenu θ sumirati poznatu vrijednost θ_3 i procijenjenu vrijednost od θ_4 . Također, kao i u prethodnom slučaju, ako postoji veza između $g(x)$ i $h(x)$ moguće je koristiti negativnu korelaciju pojedinačnih procjena za smanjenje varijance ukupne procjene.

5 Markov Chain Monte Carlo

U ovom ćemo poglavlju predstaviti metodu za generiranje uzoraka iz proizvoljne distribucije, zvanu *Markov Chain Monte Carlo*⁶. Začetnik MCMC metode je američki fizičar Nicholas Metropolis, a počiva na ideji generiranja Markovljevog lanca čija je granična distribucija jednaka željenoj ciljanoj distribuciji. Postoje brojne izmjene i poboljšanja originalnog algoritma, prije svega od strane kanadskog matematičara Hastingsa. U današnje vrijeme, svaki postupak stvaranja Markovljevog lanca čija je stacionarna distribucija jednaka ciljanoj distribuciji navodi se kao MCMC. Najznačajniji MCMC algoritam je *Metropolis-Hastings*.

Zamislimo da imamo distribuciju (moguće u velikim dimenzijama) iz koje želimo generirati uzorke ili naći očekivanu vrijednost. Osnovna ideja MCMC je da izvedemo "slučajnu šetnju" kroz distribuciju tako da dajemo prednost vrijednostima iz distribucije koje imaju veću vjerojatnost ostvarivanja. To jest, ako imamo početnu poziciju, proizvoljno odaberemo obližnju poziciju i izračunamo njenu vjerojatnost. Ako je vjerojatnost odabrane pozicije veća od vjerojatnosti pozicije iz koje smo krenuli, pomaknemo se tamo. U suprotnom, ostanemo na mjestu ili se pomaknemo na tu poziciju uz određenu vjerojatnost, o kojoj će biti govora u nastavku. Zanimljivo je da, ukoliko ponovimo ovaj postupak puno puta, svaku vrijednost (poziciju) distribucije ćemo posjetiti s frekvencijom proporcionalnom vjerojatnosti te vrijednosti.

5.1 Markovljevi lanci

Budući da su Markovljevi lanci važan dio predstojeće analize u radu, ukratko ćemo se podsjetiti važnijih pojmova i svojstava vezanih za njih. Koristit ćemo ih, kako u algoritamskom dijelu, tako i radi lakšeg interpretiranja kasnijih primjera .

Definicija 5.1. *Slučajni proces je familija slučajnih varijabli $(X_t, t \in T_0)$ na istom vjerojatnosnom prostoru (Ω, \mathcal{F}, P) , gdje je $T_0 \subseteq \mathbb{R}$.*

Uz slučajni proces $(X_t, t \in T_0)$ vežemo dva bitna skupa, skup stanja - skup svih mogućih realizacija slučajnih varijabli X_t i skup indeksa T_0 .

⁶MCMC - Markov Chain Monte Carlo

Definicija 5.2. Za slučajni proces $(X_t, t \in T)$ kažemo da je Markovljev ako vrijedi da je

$$P(a < X_t \leq b | X_{t_1} = x_1, \dots, X_{t_n} = x_n) = P(a < X_t \leq b | X_{t_n} = x_n),$$

čim je $t_1 < t_2 < \dots < t_n < t$, i uvjetne su vjerojatnosti dobro definirane.

Proces $(X_t, t \in T_0)$ je Markovljev ako za svaki $s > 0$ i t vrijedi

$$(X_{t+s} | X_u, u \leq t) \sim (X_{t+s} | X_t).$$

To je proces sa svojstvom da uz danu vrijednost procesa u trenutku t , X_t , vrijednost od X_s , $s > t$, ne ovisi o vrijednostima X_u , za $u < t$. Drugim riječima, uvjetna distribucija buduće varijable X_{t+s} , u odnosu na čitavu prošlost procesa $(X_u, u \leq t)$, jednaka je uvjetnoj distribuciji od X_{t+s} u odnosu samo na trenutnu vrijednost X_t . Ovisno o skupu stanja i skupu indeksa, Markovljev proces može se pojaviti u različitim oblicima. Markovljev proces s diskretnim skupom stanja naziva se *Markovljev lanac* (ML).

Definicija 5.3. Neka je S diskretan skup. Slučajni proces $X = (X_t, t \in T)$ na vjerojatnosnom prostoru (Ω, \mathcal{F}, P) s vrijednostima u skupu S je Markovljev lanac ako vrijedi

$$P(X_t = i | X_{t_1} = i_1, \dots, X_{t_n} = i_n) = P(X_t = i | X_{t_n} = i_n), \quad (5.1)$$

$\forall t_1, \dots, t_n, t \in T$ takve da je $t_1 < \dots < t_n < t$ i $\forall i, i_1, \dots, i_n \in S$ za koje su obje uvjetne vjerojatnosti dobro definirane.

Izraz (5.1) naziva se Markovljevo svojstvo (MS), a govori kako je ponašanje ML u neposrednoj budućnosti, uvjetno na sadašnjost i prošlost, jednako ponašanju ML u neposrednoj budućnosti uvjetovano samo na sadašnjost.

Definicija 5.4. Funkcija prijelaznih vjerojatnosti Markovljevog lanca dana je izrazom

$$p(i, s; j, t) = P(X_t = j | X_s = i), \quad s < t, i, j \in S.$$

Tako je, na primjer, funkcija prijelazne vjerojatnosti ML u jednom koraku

$$p(i, n; j, n+1) = P(X_{n+1} = j | X_n = i), \quad i, j \in S, n \in \mathbb{N}_0.$$

Teorem 5.1. *ML je u potpunosti određen poznavanjem distribucije od X_0 i funkcije prijelazne vjerojatnosti u jednom koraku $p(i, n; j, n + 1)$.*

Dokaz. Neka je $\lambda = (\lambda_i, i \in S)$ distribucija od X_0 , to jest, $P(X_0 = i), i \in S$. Zanima nas vjerojatnost $P(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n)$. Vektor (X_0, \dots, X_n) je $(n + 1)$ -dimenzionalan vektor s komponentama iz Markovljevog lanca $X = (X_n, n \in \mathbb{N})$.

Distribucija tog vektora je jedna konačnodimenzionalna distribucija ML.

Imamo sljedeće:

$$\begin{aligned}
 P(X_0 = i_0, \dots, X_n = i_n) &= P(X_n = i_n | X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \cdot \\
 &P(X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \\
 &= P(X_n = i_n | X_{n-1} = i_{n-1}) \cdot \\
 &P(X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \\
 &= p(i_{n-1}, n - 1; i_n, n) \cdot \\
 &P(X_{n-1} = i_{n-1} | X_0 = i_0, \dots, X_{n-2} = i_{n-2}) \cdot \\
 &P(X_0 = i_0, \dots, X_{n-2} = i_{n-2}) \\
 &= \dots \\
 &= p(i_{n-1}, n - 1, i_n, n) \cdots p(i_0, 0; i_1, 1) \cdot \lambda_{i_0}
 \end{aligned}$$

Konačnodimenzionalne distribucije ML su u potpunosti poznate ako je poznata distribucije od X_0 i funkcija prijelaznih vjerojatnosti u jednom koraku, a to znači da je ML u potpunosti određen distribucijom λ i funkcijom $p(i, n; j, n + 1)$. \square

Ukoliko funkcija prijelaznih vjerojatnosti u jednom koraku ne ovisi o n , to jest, za sve $n, m \in \mathbb{N}$ vrijedi

$$p(i, n; j, n + 1) = p(i, m; j, m + 1),$$

kažemo da se radi o homogenom Markovljevom lancu. Budući u ovom slučaju funkcija prijelaznih vjerojatnosti u jednom koraku ovisi samo o sadašnjem stanju i i budućem stanju j Markovljevog lanca, pišemo:

$$p(i, n; j, n + 1) = p_{ij}, \quad i, j \in S.$$

Matrica $\Pi = [p_{ij}]_{i,j \in S}$ zove se matrica prijelaznih vjerojatnosti homogenog Markovljevog lanca. Elementi ove matrice su nenegativni, a zbroj elemenata u svakom njezinom retku jednak je jedan. Matricu čiji elementi zadovoljavaju navedena svojstva nazivamo stohastičkom matricom.

Neka je $\lambda = (\lambda_i, i \in S)$ distribucija na S te neka je $\Pi = (p_{ij}, i, j \in S)$ stohastička matrica. Slučajni proces $X = (X_n, n \geq 0)$ definiran na vjerojatnosnom prostoru (Ω, \mathcal{F}, P) sa skupom stanja S je homogen Markovljev lanac s početnom distribucijom λ i prijelaznom matricom Π ako vrijedi

- (i) $P(X_0 = i) = \lambda_i, \forall i \in S$ te
- (ii) $P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = p_{ij},$
 $\forall n \geq 0$ i za sve $i_0, \dots, i_{n-1}, i, j \in S$.

Takav Markovljev lanac kraće zapisujemo (λ, Π) -Markovljev lanac.

Promotrimo jednodimenzionalne distribucije ML pomoću početne distribucije $\lambda = (\lambda_i, i \in S)$ i $\Pi = [p_{ij}]_{i,j \in S}$:

- $P(X_0 = i) = \lambda_i, \forall i \in S$
- $P(X_1 = i) = P(X_1 = i, X_0 \in S) = \sum_{i_0 \in S} P(X_1 = i, x_0 = i_0) =$
 $\sum_{i_0 \in S} P(x_0 = i_0) \cdot P(X_1 = i | x_0 = i_0) = \sum_{i_0 \in S} \lambda_{i_0} \cdot p_{i_0 i}$
- \vdots
- $P(X_n = i) = \sum_{i_0 \in S} \cdots \sum_{i_{n-1} \in S} \lambda_{i_0} p_{i_0 i_1} \cdots p_{i_{n-1} i}, i \in S.$

Iz toga slijedi

$$P(X_n = i) = (\lambda \Pi^n)_i, \quad i \in S.$$

Prijelazne vjerojatnosti ML u više koraka, to jest, $P(X_{n+m} = j | X_n = i)$ (ako je ML homogen, ta vjerojatnost jednaka je $P(X_m = j | X_0 = i)$):

$$P(X_m = j | X_0 = i) = (\Pi^m)_{ij},$$

što ćemo označavati s $p_{ij}^{(m)}$.

Teorem 5.2 (Chapman-Kolmogorovljeve jednakosti). *Prijelazne vjerojatnosti homogenog (λ, Π) -ML zadovoljavaju Chapman-Kolmogorovljeve jednakosti:*

$$p_{ij}^{(m)} = \sum_{k \in S} p_{ik}^{(r)} \cdot p_{kj}^{(m-r)},$$

$\forall r \in \{1, \dots, m-1\}, i, j \in S$, to jest,

$$\Pi^m = \Pi^r \Pi^{m-r}.$$

Slučajni proces $(X_t, t \in T)$ je stacionaran u užem smislu ili strogo stacionaran ako su distribucije slučajnih vektora $(X_{t_1}, \dots, X_{t_n})$ i $(X_{t_1+h}, \dots, X_{t_n+h})$, za proizvoljan $h > 0$, jednake za proizvoljne $t_1, \dots, t_n \in T$.

Slučajni proces $(X_t, t \in T)$ je stacionaran u širem smislu ili slabo stacionaran ako su ispunjena sljedeća tri zahtjeva:

- (1) $E[X_t^2] < \infty, \forall t \in T$
- (2) $E[X_t] = c, \forall t \in T, c \in \mathbb{R}$
- (3) $Cov(X_t, X_{t+h})$ ovisi samo o h , za proizvoljan $t \in T$, to jest
 $Cov(X_t, X_{t+h}) = f(h)$

Strogo stacionaran proces s konačnim drugim momentima je ujedno i slabo stacionaran. Stacionarnost kod slučajnih procesa znači da se vjerojatnosna svojstva procesa ne mijenjaju kroz vrijeme. Preciznije, ako je $X = (X_n, n \geq 0)$ stacionaran slučajni proces, tada je distribucija svih slučajnih varijabli X_n jednaka. Prethodnom tvrdnjom smo zapravo karakterizirali strogu stacionarnost.

Definicija 5.5. *Slučajni proces $X = (X_n, n \geq 0)$ definiran na vjerojatnosnom prostoru (Ω, \mathcal{F}, P) je strogo stacionaran ako za sve $k \geq 0$ i sve $n \geq 0$, slučajni vektori (X_0, X_1, \dots, X_k) i $(X_n, X_{n+1}, \dots, X_{n+k})$ imaju istu distribuciju (u odnosu na vjerojatnost P).*

Specijalno, ako X poprima vrijednosti u prebrojivom skupu stanju S , tada uzimajući $k = 0$ slijedi $P(X_n = i) = P(X_0 = i)$ za sve $i \in S$ i sva vremena $n \geq 1$. Dakle, kao specijalan slučaj dobivamo da se jednodimenzionalne distribucije ne mijenjaju kroz vrijeme.

Definicija 5.6. *Neka je $X = (X_n, n \geq 0)$ Markovljev lanac s prebrojivim skupom stanja S i prijelaznom matricom Π . Vjerojatnosna distribucija $\pi = (\pi_i, i \in S)$ na S je stacionarna distribucija (ili invarijantna distribucija) Markovljevog lanca X (odnosno prijelazne matrice Π) ako vrijedi*

$$\pi = \pi \Pi,$$

odnosno po komponentama

$$\pi_j = \sum_{k \in S} \pi_k p_{kj}, \forall j \in S,$$

distribucija sa svojstvom da ako iz nje starta, Markovljev lanac je strogo stacionaran proces.

Definicija 5.7. Neka je $X = (X_n, n \geq 0)$ Markovljev lanac na skupu stanja S i prijelaznom matricom Π . Vjerojatnosna distribucija $\pi = (\pi_i, i \in S)$ naziva se granična distribucija Markovljevog lanca X (odnosno prijelazne matrice Π) ako za sve $i, j \in S$, vrijedi

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j.$$

Mogu nas zanimati vjerojatnosti da će se sustav kojeg promatramo nalaziti u određenom stanju u nekom budućem vremenu. Te će vjerojatnosti obično ovisiti o kojem vremenu u budućnosti je riječ. U mnogim Markovljevim lancima vjerojatnost pojedinog stanja približavat će se graničnoj vrijednosti kako vrijeme bude teklo prema beskonačnosti. Drugim riječima, u dalekoj budućnosti vjerojatnosti se neće puno mijenjati iz jednog prijelaza u drugi. Spomenute granične vrijednosti nazivaju se granične vjerojatnosti.

Stacionarnu i graničnu distribuciju povezujemo sljedećem rezultatom. Neka je S konačan skup stanja, te pretpostavimo da za neki $i \in S$,

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j,$$

za sve $j \in S$. Tada je $\pi = (\pi_j, j \in S)$ stacionarna distribucija.

Općenito, granična distribucija ne mora postojati iako postoji (jedinstvena) stacionarna distribucija. Pretpostavimo da imamo $X = (X_n : n \geq 0)$ Markovljev lanac sa skupom stanja $S = 1, 2$ i prijelaznom matricom

$$\Pi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Rješavanjem sustava $\pi = \pi\Pi$ dobivamo $\pi = (\pi_1, \pi_2) = (\frac{1}{2}, \frac{1}{2})$, pa vidimo da postoji jedinstvena stacionarna distribucija Markovljevog lanca. No, kako je $\Pi^{2n} = I$ i $\Pi^{2n+1} = \Pi$, $\forall n \in N$, očito je da $\lim_{n \rightarrow \infty} p_{ij}^{(n)}$ ne postoji niti za jedan par $i, j \in S$, pa prema tome, ovaj Markovljev lanac nema graničnu

distribuciju. Bitnu ulogu u ovom primjeru ima svojstvo periodičnosti Markovljevog lanca, to jest činjenica da je $p_{ii}^n > 0$ samo za parne n , odnosno da svako stanje ima period 2.

Za razumijevanje razvoja Markovljevog lanca važno je znati koji su putevi kroz prostor stanja mogući. Kao najjednostavnije postavlja se pitanje koja stanja lanac uopće može posjetiti krenuvši iz nekog zadanog stanja.

Za Markovljev lanac $X = (X_n, n \geq 0)$ sa skupom stanja S definiramo *prvo vrijeme pogađanja* skupa $B \subset S$ kao

$$T_B = \min\{n \geq 0 : X_n \in B\}.$$

Definicija 5.8. Za stanja $i, j \in S$ kažemo da je j *dostižno* iz i , u oznaci $i \rightarrow j$, ako vrijedi da je

$$P(T_j < \infty | X_0 = i) > 0.$$

U prijevodu, stanje j dostižno je iz stanja i ako lanac s vjerojatnošću u konačnom vremenu posjeti j krenuvši iz i .

Definicija 5.9. Stanja $i, j \in S$ *komuniciraju*, u oznaci $i \longleftrightarrow j$, ako vrijedi $i \rightarrow j$ i $j \rightarrow i$.

Relacija komuniciranja je relacija ekvivalencije na $S \times S$, te stoga inducira particiju prostora stanja S na klase.

Definicija 5.10. Markovljev lanac X je *ireducibilan* ako se prostor stanja S sastoji samo od jedne klase komuniciranja.

Definicija 5.11. Neka je X Markovljev lanac s prijelaznom matricom Π . Za stanje $i \in S$, označimo s $d(i)$ najveći zajednički djelitelj skupa $\{n \geq 1 : p_{ii}^{(n)} \geq 0\}$, gdje je $d(i) = 1$ ako je taj skup prazan. Kažemo da je stanje i *aperiodično*, ako je $d(i) = 1$. U suprotnom je i *periodičko stanje*, a $d(i)$ se zove *period* od i .

Osnovni rezultat o postojanju granične distribucije dan je sljedećim teoremom, kojeg navodimo bez dokaza. [5]

Teorem 5.3. Neka je λ proizvoljna vjerojatnosna distribucija na skupu stanja S . Pretpostavimo da je $X = (X_n, n \geq 0)$ (λ, Π) -Markovljev lanac koji je ireducibilan i aperiodičan, te ima stacionarnu distribuciju π . Tada je

$$\lim_{n \rightarrow \infty} P(X_n = j) = \pi_j, \forall j \in S.$$

Specijalno,

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j, \forall i, j \in S,$$

to jest, stacionarna distribucija ujedno je i granična.

Definicija 5.12. Za ireducibilan (λ, Π) -Markovljev lanac $X = (X_n, n \geq 0)$ kažemo da je reverzibilan, ako je za sve $N \geq 1$, $(X_{N-n}, 0 \leq n \leq N)$ ponovno (λ, Π) -Markovljev lanac, gdje je λ distribucija na S .

5.2 Metropolis-Hastings algoritam

Osnovna ideja algoritma je simulacija Markovljevog lanca takva da se njegova stacionarna distribucija podudara s ciljanom distribucijom.

Za motivaciju, pretpostavimo da želimo generirati slučajnu varijablu X uzimajući vrijednosti iz $\mathcal{X} = \{1, \dots, m\}$ u skladu s ciljanom distribucijom $\{\pi_i\}$, gdje je

$$\pi_i = \frac{b_i}{C}, \quad i \in \mathcal{X},$$

gdje pretpostavljamo da su svi $\{b_i\}$ strogo veći od 0, m je velik, a da je konstantu $C = \sum_{i=1}^m b_i$ teško izračunati. Zatim konstruiramo Markovljev lanac $\{X_t, t = 0, 1, \dots\}$ na \mathcal{X} čiji se razvoj oslanja na proizvoljnu matricu prijelaza $Q = [q_{ij}]$ na sljedeći način:

- Kada je $X_t = i$, generiraj slučajnu varijablu Y zadovoljavajući $P(Y = j) = q_{ij}, j \in \mathcal{X}$. Na taj način, Y je generirana iz distribucije na m -teročlanom skupu dane i -tim retkom iz Q .

- Ako je $Y = j$, stavi

$$X_{t+1} = \begin{cases} j, & \text{s vjerojatnošću } \alpha_{ij} = \min \left\{ \frac{\pi_j q_{ji}}{\pi_i q_{ij}}, 1 \right\} = \min \left\{ \frac{b_j q_{ji}}{b_i q_{ij}}, 1 \right\} \\ i, & \text{s vjerojatnošću } 1 - \alpha_{ij}. \end{cases}$$

Slijedi da $\{X_t, t = 0, 1, \dots\}$ ima funkciju prijelazne vjerojatnosti u jednom koraku $\Pi = (p_{ij})$ danu s

$$p_{ij} = \begin{cases} q_{ij} \alpha_{ij}, & i \neq j, \\ 1 - \sum_{k \neq i} q_{ik} \alpha_{ik}, & i = j. \end{cases}$$

Tada, uz ranije definirani α_{ij} , vrijedi

$$\pi_i p_{ij} = \pi_j p_{ji}, \quad i, j \in \mathcal{X}.$$

Drugim riječima, Markovljev lanac je reverzibilan i ima stacionarnu distribuciju $\{\pi_i\}$. K tomu, stacionarna distribucija je također i granična distribucija ako je Markovljev lanac ireducibilan i aperiodičan.

Za proširenje prethodnog MCMC pristupa kako bi se generirao uzorak iz proizvoljne višedimenzionalne funkcije gustoće $f(x)$ (umjesto π_i), na mjesto q_{ij} dolazi nenegativna funkcija prijelaza $q(x, y)$, koju nazivamo *predložena* funkcija. Gledajući tu funkciju kao uvjetnu funkciju gustoće, možemo pisati $q(y|x)$ umjesto $q(x, y)$. Vjerojatnost $\alpha(x, y)$ zove se *vjerojatnost prihvatanja*. Izvorni Metropolis algoritam bio je osmišljen za simetrične predložene funkcije, to jest, za $q(x, y) = q(y, x)$, a zatim je Hastingsova modifikacija proširila uporabu i za nesimetrične predložene funkcije. Takav algoritam nazivamo Metropolis-Hastings algoritam, a odgovarajući Markovljev lanac zovemo Metropolis-Hastings Markovljev lanac. Primjetimo kako se, slično kao i metoda prihvatanja-odbijanja, ovaj algoritam zasniva na strategiji pokušaja i pogreške.

Algoritam 5.1 (Metropolis-Hastings algoritam).

Neka je X_t početno stanje:

1. Generiraj $Y \sim q(X_t, y)$.
2. Generiraj $U \sim \mathcal{U}(0, 1)$ i vrati

$$X_{t+1} = \begin{cases} Y, & U \leq \alpha(X_t, Y) \\ X_t, & \text{inače,} \end{cases} \quad (5.2)$$

gdje je

$$\alpha(x, y) = \min\{\rho(x, y), 1\}, \quad (5.3)$$

uz

$$\rho(x, y) = \frac{f(y) q(y, x)}{f(x) q(x, y)}. \quad (5.4)$$

Ponavljajući korake 1 i 2, dobivamo niz X_1, X_2, \dots zavisnih slučajnih varijabli, pri čemu je X_t približno distribuirana u skladu s $f(x)$, za velike t .

U koraku 1 vidimo kako predložena distribucija generira vrijednost y , koja ovisi samo o tome gdje smo bili ranije (x_t), zbog čega je jasno zašto je riječ o Markovljevom lancu, budući da je svako novo odabrano stanje ovisno samo o stanju u kojem smo bili neposredno prije. U izrazu (5.4), prvi faktor uspoređuje vjerojatnosti predloženog stanja i stanja u kojem smo trenutno, dok se u brojniku drugog faktora nalazi vjerojatnost stanja u kojem smo trenutno uz dano predloženo stanje, a u nazivniku vjerojatnost pomaka na predloženo stanje uz dano trenutno stanje. U već spomenutom izvornom Metropolis algoritmu, taj izraz nestaje, budući da se u tom slučaju odabire takva funkcija q koja stvara simetričnu slučajnu šetnju.

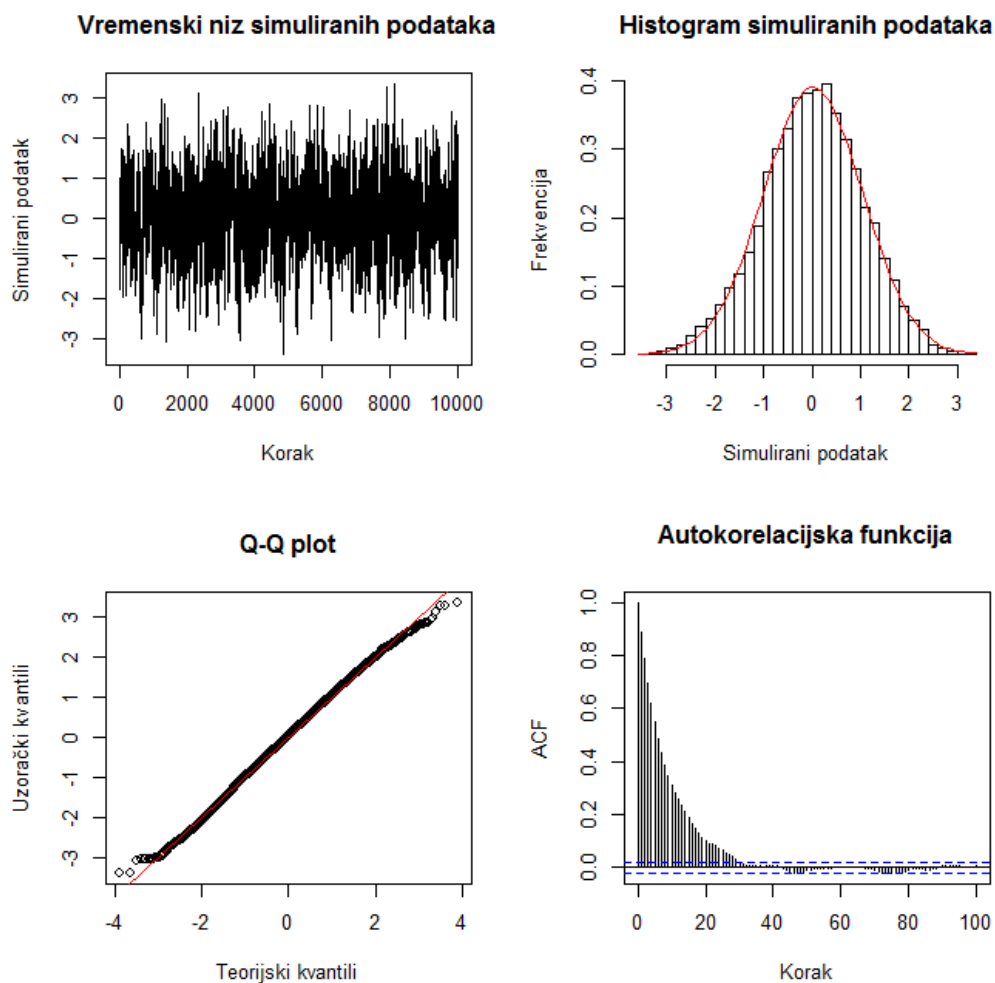
Zamislimo da je predloženo stanje y vjerojatnije od stanja u kojem smo trenutno. To znači da (5.3) iznosi 1, to jest, U će uvijek biti manje od $\alpha(X_t, Y)$, što ćemo prihvatiti i pomaknuti se na predloženu poziciju. U slučaju kada je trenutno stanje vjerojatnije od predloženoga, izraz (5.4) je manji od 1, ali i dalje postoji neka vjerojatnost da vrijednost U bude manja od $\alpha(X_t, Y)$, pa opet prihvaćamo, to jest, $X_{t+1} = Y$.

5.2.1 Primjeri

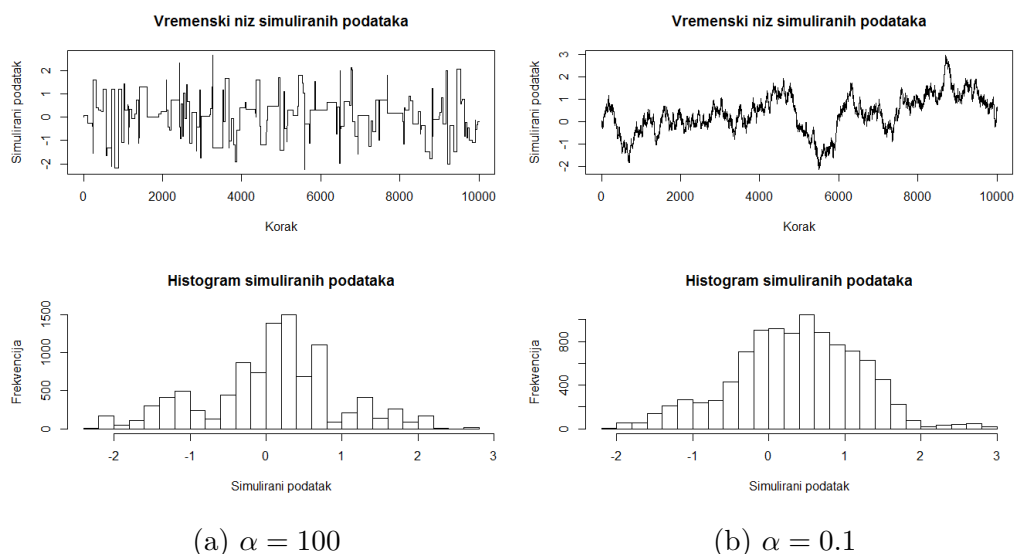
U ovom ćemo potpoglavlju prikazati rad prethodnih algoritama na konkretnim situacijama. Koristili smo programski jezik R.

Primjer 5.1. *Metropolis algoritam*

Simulirajmo podatke iz normalne distribucije s očekivanjem 0 i varijancom 1, uz predloženu uniformnu distribuciju. Lanac polazi iz 0, a u svakom koraku generira se slučajni broj iz $\mathcal{U}(-\alpha, \alpha)$. Odluka o tome hoće li kandidat biti prihvaćen donosi se na temelju usporedbe je li vjerojatnost prema $\mathcal{U}(0, 1)$ distribuciji manja od vjerojatnosti prihvaćanja. Slika 5 prikazuje dobro "izmješani" lanac i intuitivno prihvatljivu normalnu distribuiranost podataka. Međutim, ovo je rezultat izbora $\alpha = 1$. Druge vrijednosti alpha neće utjecati na stacionarnu distribuciju, ali će utjecati na brzinu promjene vrijednosti lanca.

Slika 5: Granična $\mathcal{N}(0,1)$ distribucija

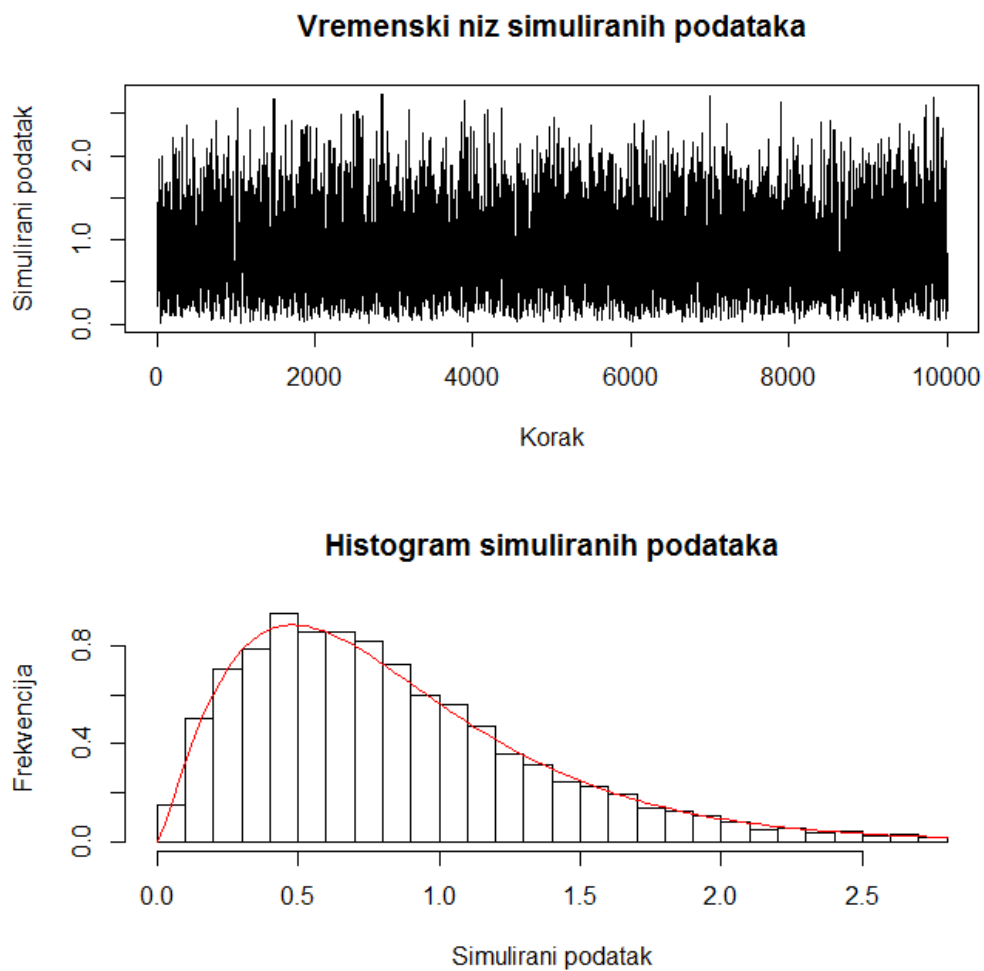
Slika 6 rezultat je odabira $\alpha = 100$ i $\alpha = 0.1$. $\alpha = 100$ daje "preosjetljiv" lanac, u smislu da je jako malo kandidata prihvaćeno, ali kada su prihvaćeni predstavljaju vrlo značajan pomak unutar uzorka. $\alpha = 0.1$ stvara lanac kojeg možemo smatrati "nekritičnim", u smislu da je većina kandidata prihvaćena, ali predstavljaju prilično konzervativne i jednostavne pomake u lancu.



Slika 6: Vremenski niz i histogram simuliranih podataka u ovisnosti o vrijednosti α

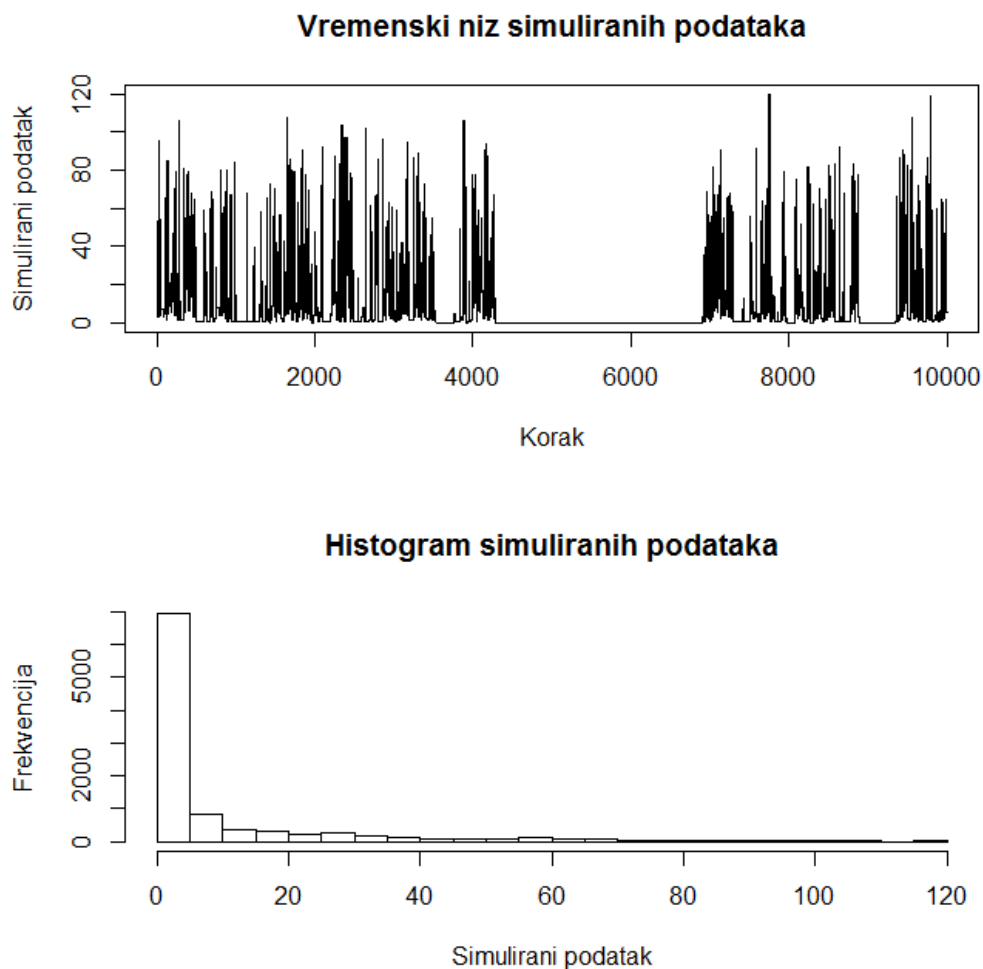
Primjer 5.2. Metropolis-Hastings algoritam

Simulirajmo podatke iz gama distribucije proizvoljnih parametara uz predloženu normalnu distribuciju s jednakim očekivanjem i varijancom kao i kod željene gama distribucije. Lanac polazi iz 0, a u svakom stupnju predlaže se kandidat iz $\mathcal{N}(a/b, a/(b*b))$. Odluka o tome hoće li kandidat biti prihvaćen donosi se na temelju usporedbe je li vjerojatnost prema $\mathcal{U}(0, 1)$ distribuciji manja od vjerojatnosti prihvaćanja. Slika 7 prikazuje dobro "izmještan" lanac i intuitivno prihvatljivu gama distribuiranost podataka.



Slika 7: Vremenski niz i histogram simuliranih podataka za $a = 2.3$, $b = 2.7$

Ukoliko bi promijenili vrijednosti parametara te uzeli, na primjer, $a = 0.1$ i $b = 0.01$, vidimo kako bi se pojavio problem "zapinjanja" uzorka na vrlo malim i vrlo velikim vrijednostima, što je prikazano slikom 8.

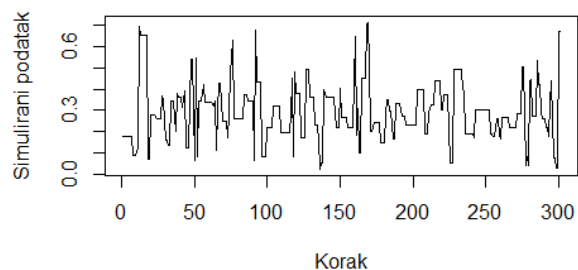


Slika 8: Vremenski niz i histogram simuliranih podataka za $a = 0.1$, $b = 0.01$

Primjer 5.3.

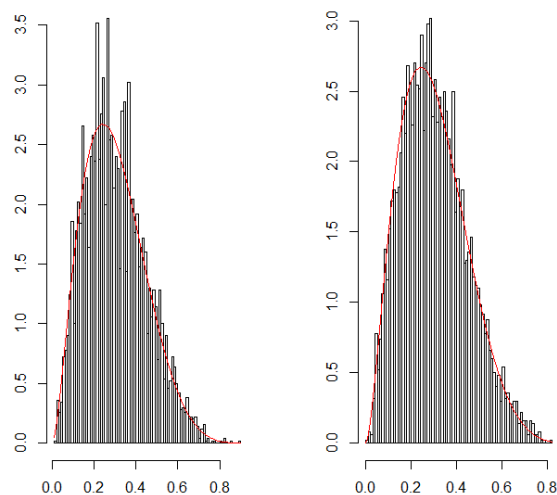
Pokušajmo iskoristiti Metropolis-Hastings algoritam za rješavanje primjera 3.4, u kojem je ciljana gustoća f bila $\mathcal{Be}(2.7, 6.3)$, a predložena funkcija $\mathcal{U}(0, 1)$.

Ako поближе pogledamo lanac u određenom vremenskom periodu (slika 9, konkretno, $4500 \leq t \leq 4800$, od ukupno 5000 generiranih vrijednosti) uočavamo karakterističnu osobinu Metropolis-Hastings niza, a to je da se, za neke vremenske intervale, niz ne mijenja, budući da su odgovarajući predloženi kandidati odbačeni.



Slika 9: Niz za $t = 4500, \dots, 4800$ simuliran Metropolis-Hastings algoritmom

No, promatrajući čitav lanac, vidimo da histogram dobro aproksimira ciljane $\mathcal{Be}(2.7, 6.3)$ (slika 10). Lijevi dio prikazuje histogram uzorka dobivenog algoritmom, a desni dio uzorak dobiven ugrađenom R naredbom za generiranje iz beta distribucije. Provjeru vršimo ranije spomenutim Kolmogorov-Smirnovljevim testom, prilikom čega dobivamo p -vrijednost 0.4209, te nemamo razloga odbaciti nultu hipotezu o jednakosti distribucije uzorka.



Slika 10: Histogram $\mathcal{Be}(2.7, 6.3)$ slučajnih varijabli

Literatura

- [1] James E. Gentle, Random Number Generation and Monte Carlo Methods, Springer
- [2] Reuven Y. Rubinstein i Dirk P. Kroese, Simulation and the Monte Carlo Method, Wiley
- [3] Paul Glasserman, Monte Carlo Methods in Financial Engineering, Springer
- [4] Pierre Bremaud, Markov Chains; Gibbs Fields, Monte Carlo Simulation and Queues, Springer
- [5] Christian P. Robert i George Casella, Introducing Monte Carlo Methods with R, Springer
- [6] Zoran Vondraček, Markovljevi lanci, skripta, Sveučilište u Zagrebu, Odjel za matematiku
- [7] Mirta Benšić i Nenad Šuvak, Uvod u vjerojatnost i statistiku, Sveučilište J.J. Strossmayera, Odjel za matematiku

Sažetak

Monte Carlo metode široka su klasa algoritama koji se oslanjaju na operativno slučajno uzorkovanje u svrhu dobivanja numeričkih rezultata. Najpogodnije su za uporabu u situacijama kada je teško ili nemoguće dobiti procjenu nekog izraza u konačnom broju koraka. Tehnika je to koja može biti korištena u posve različitim područjima, poput financija, energetike i mnogih drugih.

U jezgri Monte Carlo simulacija nalazi se generiranje slučajnih brojeva, umjesto kojih mnoge aplikacije koriste pseudoslučajne brojeve.

U prvoj cjelini rada predstavili smo tehnike za generiranje slučajnih brojeva i slučajnih varijabli. Zatim smo se upoznali s Monte Carlo metodom u svom najopćenitijem obliku. Na kraju, analizirali smo Monte Carlo metode koje se baziraju na generiranju Markovljevog lanca čija je granična distribucija jednaka ciljanoj distribuciji.

Ključne riječi: Generiranje slučajnih brojeva, uniformna distribucija, linearni kongruentni generator, generiranje slučajnih varijabli, metoda inverzne transformacije, metoda prihvatanja-odbijanja, Monte Carlo metode, Markov Chain Monte Carlo, Metropolis-Hasting algoritam

Title and summary

Monte Carlo Simulation

Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. They are most useful when it is difficult or impossible to obtain a mathematical expression that can be evaluated in a finite number of operations. The technique is used in such widely disparate fields as finance, energy, and many more.

At the kernel of Monte Carlo simulation is random number generation, instead of which, most applications use pseudorandom numbers.

In the first part of this paper, we introduced techniques for random number and random variable generation. Then, we presented Monte Carlo simulation in its simplest form. At last, we analyzed Monte Carlo methods which rely on the idea of generating a Markov chain whose limiting distribution is equal to the desired target distribution.

Key words: Random number generation, uniform distribution, linear congruential generator, random variable generation, inverse-transform method, acceptance-rejection method, Monte Carlo methods, Markov Chain Monte Carlo, Metropolis-Hastings algorithm

Životopis

Rođen sam 16. rujna 1990. godine u Osijeku gdje sam završio osnovnu školu i Prirodoslovno-matematičku gimnaziju. Obrazovanje sam nastavio na Pred-diplomskom studiju matematike na Odjelu za matematiku u Osijeku 2009. godine koji sam završio 2012. godine te stekao naziv Sveučilišnog prvostupnika (baccalaureus) matematike. Nastavio sam studiranje na Sveučilišnom diplomskom studiju matematike - smjer Financijska matematika i statistika.