Monte Carlo metode

Budetić, Mia

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Department of Mathematics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:126:012597

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Diplomski studij matematike
Financijska matematika i statistika

Mia Budetić

Monte Carlo metode

Diplomski rad

Osijek, 2020.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Diplomski studij matematike
Financijska matematika i statistika

Mia Budetić

Monte Carlo metode

Diplomski rad

Mentor: prof. dr. sc. Mirta Benšić

Osijek, 2020.

Sadržaj

Uvod

Monte Carlo metode su bilo koji matematički model i algoritam čija je glavna značajka korištenje velikog broja slučajnih brojeva u rješavanju različitih problema. Metode, odnosno simulacije se najčešće koriste za dobivanje numeričkih rješenja kod problema koje je vrlo teško riješiti analitički.

Prvotno su se Monte Carlo metode u statistici nazivale „model uzorkovanja“ i koristile su se za provjeru svojstava procjene. Prvo dokumentirano pojavljivanje Monte Carlo aproksimacije može se pronaći u nekim publikacijama matematičara Comte de Buffona[1] početkom 18. stoljeća. Monte Carlo uzorkovanje počelo je biti puno poznatije 1940-ih i početkom 1950-ih gdje je služilo za rješavanje problema u fizici koji su se odnosili na razvoj atomskog oružja. Metodu je 1946. godine osmislio Stanisław Ulam[2] dok je radio na razvoju nuklearnog oružja u Los Almos National Laboratory-u. Vrijednost metode ubrzo je prepoznao John von Neumann[3] pišući program za prvo računalo $ENIAC$[4], koji je probleme neutronske difuzije u fizibilnim materijalima rješavalo upravo Monte Carlo metodom. Rad Ulama i von Neumanna zahtijevao je tajno ime pa je naziv „Monte-Carlo metoda” dobiveno po gradu u državi Monaco, poznatom po kockarnicama u kojima je ujak Ulama često kockao.

Zbog velikog broja matematičkih operacija i ponavljanja istih, Monte Carlo metode ulaze u široku upotrebu tek s naglim razvojem računala kada se počinju detaljnije proučavati. Metodologija simulacije oslanja se na dobar izvor brojeva koji se čine nasumičnim. Ti brojevi moraju proći mnoge statističke testove da bi se potvrdilo kako su u skladu sa slučajnim nizom. Metode kreiranja slučajnih brojeva u simulaciju uzoraka iz različitih distribucija su i danas među važnijim temama u statistici.

Monte Carlo simulacije usko su povezane s proučavanjem šanse za pobjedu, odnosno gubitak, u igrama na sreću. Također, postale su jedno od najvažnijih oruđa na svim područjima znanosti, od numeričke matematike, fizikalne kemije, statističke fizike, računalne grafike i financija.

U radu je u prvom dijelu najprije opisan uvodni primjer, poslije čega su definirani osnovni pojmovi i uvedena terminologija koja se koristi u daljnjem radu. Zatim su analizirani načini generiranja nizova slučajnih brojeva čija se kvaliteta ispituje statističkim testovima. Nakon toga, dan je uvid u dvije različite metode generiranja realizacija slučajnih varijabli, metode inverzije i metode odbacivanja. U zadnjem poglavlju opisane su Monte Carlo metode i njihove primjene. Primjeri obrađenih algoritama predstavljeni su kroz implementaciju u programskom jeziku R.

[1] Georges Louis Leclerc Comte de Buffon(1707.-1788.) francuski prirodoslovac, matematičar, kozmolog i enciklopedist

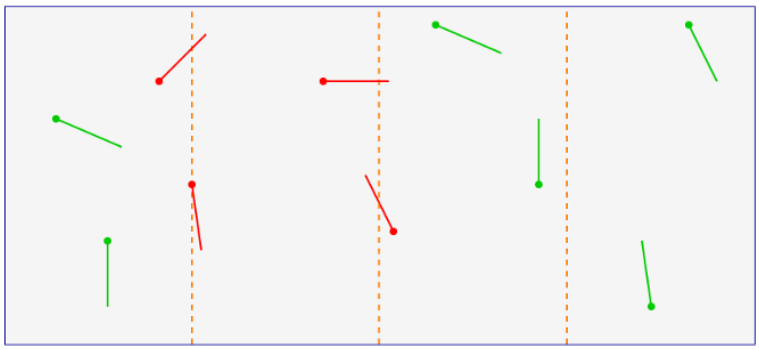
[2] Stanisław Marcin Ulam(1909.-1984.) poljski matematičar i nuklearni fizičar

[3] John von Neumann(1903.-1957.) mađarsko-američki matematičar, fizičar, informatičar, inžinjer, polimat

[4] Electronic Numerical Integrator And Computer(1946.) prvo elektroničko računalo konstruirano u Americi

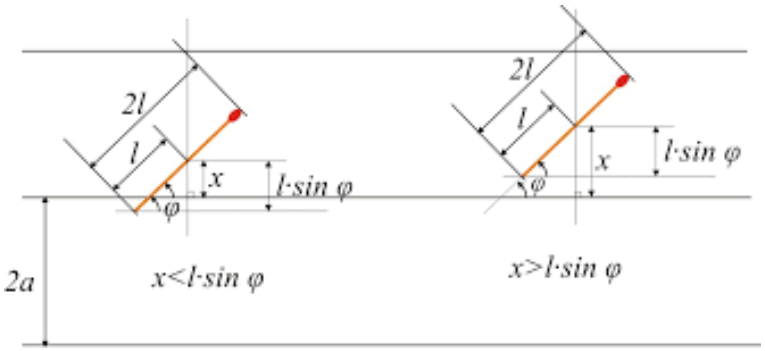
1 Počeci Monte Carlo metoda

Osnovni koncept Monte Carlo metoda potiče još iz 18. stoljeća kada je francuski znanstvenik Buffon prezentirao metodu bacanja igle kako bi izračunao broj π. Navedeni primjer smatran je najranijim i najzanimljivijim primjerom primjene Monte Carlo metoda, a poznat je pod nazivom Buffonov problem igle.



Slika 1: Buffonov problem igle

Metoda se bazira na tome da se igla duljine $2l$ nasumično baca na ravninu podijeljenu paralelnim pravcima udaljenim jedan od drugog za $2a$, pri čemu je $l < a$, što znači da igla ne može sjeći dva pravca istovremeno. Može se dokazati da je vjerojatnost da igla siječe neki od pravaca jednaka $P = \frac{2l}{\pi a}$. Naime, neka je φ - manji kut kojeg igla zatvara s pravcem i x-udaljenost središta igle do najbližeg pravca, kao što je prikazano slikom (2).



Slika 2: Položaj igle u $Buffonovom$ problemu

Kut φ može imati samo vrijednosti iz intervala $[0, \frac{\pi}{2}]$, a x iz $[0, a]$. Slijedi da je skup elementarnih događaja jednak $\Omega = [0, \frac{\pi}{2}] \times [0, a]$. Iz slike (2), vidljivo je da će igla sjeći pravac samo u slučaju kada je $x < l\ sin\ \varphi$. U situaciji kada vrijedi jednakost igla će samo dirati pravac.
Neka je A događaj koji predstavlja realizaciju da je igla presjekla pravac, odnosno

$$A = \{Igla\ siječe\ pravac\} = \{(\varphi, x) \in \Omega : x < l\ sin\ \varphi\}.$$

Primjenom geometrijske vjerojatnosti dobiju se sljedeći rezultati

$$\lambda(A) = \int_0^{\frac{\pi}{2}} l\ sin\varphi\ d\varphi = l$$

$$P = P(A) = \frac{\lambda(A)}{\lambda(\Omega)} = \frac{l}{\frac{a\pi}{2}} = \frac{2l}{a\pi}.$$

Budući da vjerojatnost može biti procijenjena i kao omjer ukupnog broja bacanja u kojemu je igla pogodila linije i ukupnog broja bacanja, vrijednost broja π se tada može računati kao $\pi = \frac{2l}{Pa}$.

Ljudi su tijekom povijesti stvarno izvodili ovaj pokus. Sam Buffon je 1777. bacio iglu 10000 puta i dobio $\pi \approx 3.15$. Jedan od najuspješnijih „bacača" bio je talijanski matematičar Mario Lazzarini koji je 1901. bacio iglu „samo" 3408 puta i dobio točnost broja π u šest znamenaka. Njegova igla bila je dugačka 2.5 cm, dok je duljina između pravaca bila 3 cm. Broj bacanja u kojima je igla pogodila pravce iznosio je 1808 pa je aproksimacija broja π dobivena iz:

$$\frac{1808}{3408} = \frac{2 \times 2.5}{3 \times \hat{\pi}} \Rightarrow \hat{\pi} = \frac{355}{113} = 3.1415929204.$$

2 Osnovni pojmovi

Prije samog postupka objašnjenja Monte Carlo metoda definirat će se neki ključni pojmovi (po uzoru na [1]) potrebni u radu, poput slučajnog vektora i njegove distribucije, zatim statističkog modela i jednostavnog slučajnog uzorka. Također će se definirani procjenitelj te iskazati centralni granični teorem i zakon velikih brojeva.

Definicija 2.1. *Neka je* $(\Omega, \mathcal{F}, P)$ *vjerojatnosni prostor pridružen slučajnom pokusu. Funkciju* $(X_1, \ldots, X_n)$ *koja svakom ishodu pokusa pridružuje uređenu n - torku realnih brojeva* $(x_1, \ldots, x_n)$ *zovemo* **n − dimenzionalan slučajni vektor** *ako vrijedi:*

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

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

Definicija 2.2. *Neka je* $(\Omega, \mathcal{F}, P)$ *vjerojatnosni prostor pridružen slučajnom pokusu i* $(X_1, \ldots, X_n)$ *slučajni vektor. Funkciju*

$$F : \mathbb{R}^n \to [0, 1],$$

$$F(x_1, \ldots, x_n) = P\{X_1 \leqslant x_1, \ldots, X_n \leqslant x_n\}$$

zovemo **funkcija distribucije slučajnog vektora** $(X_1, \ldots, X_n)$.

Zadati statistički model znači opisati poznate karakteristike slučajnog vektora za kojeg se smatra da 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 se modeliraju podaci, odnosno statistički model jest familija funkcija distribucije koja se uzima u obzir za zaključivanje u danom problemu.

Definicija 2.3. Statistički model $\mathcal{P}$ *je familija dozvoljenih funkcija distribucije slučajnog vektora za koji baza podataka čini jednu realizaciju.*

Definicija 2.4. *Statistički model zovemo model* **jednostavnog slučajnog uzorka** *iz funkcije distribucije F ako za slučajni vektor $(X_1, \ldots, X_n)$, čiju realizaciju čine podaci $(x_1, \ldots, x_n)$, vrijedi:*

1. *slučajne varijable $X_1, \ldots, X_n$ su nezavisne ,*
2. *sve slučajne varijable $X_1, \ldots, X_n$ imaju istu funkciju distribucije F.*

Definicija 2.5. *Neka je $\mathcal{P} = \{F_\theta : \theta \in \mathbf{\Theta} \subseteq \mathbb{R}^k\}$ parametarski statistički model, θ k-dimenzionalan parametar, a $\mathbf{\Theta} \subseteq \mathbb{R}^k$ prostor parametara tj. skup svih dozvoljenih vrijednosti nepoznatog parametra θ. Neka je $(X_1, \ldots, X_n)$ slučajni vektor s distribucijom iz $\mathcal{P}$ i $t : \mathbb{R}^n \to \mathbf{\Theta}$. Slučajni vektor $T = t(X_1, \ldots, X_n)$ jest* **procjenitelj** *za θ.*

Teorem 2.6. *(Kolmogorovljev jaki zakon velikih brojeva)* ([9], 416.*str*)
Neka je $(X_n,\ n \in \mathbb{N})$ niz nezavisnih jednako distribuiranih slučajnih varijabli na $(\Omega, \mathcal{F}, P)$. Tada niz $(\overline{X}_n,\ n \in \mathbb{N})$ konvergira $(g.s)$ ako i samo ako $E[X_1]$ postoji i u tom slučaju je

$$(g.s) \lim_{n\to\infty} \overline{X}_n = E[X_1].$$

Teorem 2.7. *(Levyjev centralni granični teorem)*([9], 507.*str*)
Neka je $(X_n,\ n \in \mathbb{N})$ niz nezavisnih jednako distribuiranih slučajnih varijabli na $(\Omega, \mathcal{F}, P)$ s očekivanjem μ i varijancom $\sigma^2 < \infty$. Tada

$$\frac{\overline{X}_n - \mu}{\sigma}\sqrt{n} \ \xrightarrow{D}\ \mathcal{N}(0,1),\ n \to \infty.$$

Kod modela jednostavnog slučajnog uzorka promatrana veličina smatra se slučajnom varijablom s funkcijom distribucije F, a vrijednosti varijable izmjerene na jedinkama iz uzorka nezavisnim realizacijama te slučajne varijable. U nastavku rada koristit će se termin slučajni uzorak za slučajni vektor $(X_1, \ldots, X_n)$ statističkog modela, a termin uzorak za njegovu realizaciju $(x_1, \ldots, x_n)$, to jest podatke.

Primjera radi, izračunavanje prosječne visine odraslog stanovništva neke zemlje zahtijeva mjerenje visine svake osobe koja čini tu populaciju, zbrajanje tih podataka i potom dijeljenje dobivenog broja s ukupnim brojem izmjerenih osoba. Izračunavanje nije nemoguće napraviti, no tada bi taj zadatak zahtijevao puno vremena. Ono što se može učiniti umjesto toga je uzeti neki uzorak te populacije i izračunati njegovu prosječnu visinu. Malo je vjerojatno da će se dobiti točna prosječna visina čitavog stanovništva, međutim, ova tehnika daje rezultat koji je općenito dobar približni iznos stvarnog broja. U većini slučajeva aproksimacija i točan prosjek cijelog stanovništva bit će različiti brojevi. Razlika između aproksimacije i stvarnog rezultata bit će manja povećanjem veličine uzorka.

Pretpostavit će se da je visina osobe realizacija slučajne varijable iz neke distribucije. Stoga, kada se uzorkuje populacija nasumičnim odabirom osoba iz populacije i mjerenjem njihove visine kako bi se aproksimirala prosječna visina, svaka mjera će biti jedna realizacija. Visina

populacije modelirat će se slučajnom varijablom X, a njene vrijednosti x_i predstavljat će visine ljudi koji čine tu populaciju, pri čemu će i biti oznaka za i-tog čovjeka te populacije. Koncept aproksimacije prosječne visine odrasle populacije iz nekog uzorka veličine N dana je sljedećom formulom:

$$Aproksimacija(Prosjek(X)) = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

To je u suštini ono što se naziva Monte Carlo aproksimacijom. Ukratko, Monte Carlo metodama se pomoću generiranog slučajnog uzorka dobivaju realizacije kojima se aproksimira očekivanje slučajne varijable.

3 Simulacija nezavisnih slučajnih uzoraka iz Uniformne $\boldsymbol{U}(0,1)$ distribucije

Kako bi se uopće mogle koristiti Monte Carlo metode, potreban je neki izvor slučajnosti. U prošlosti su taj izvor predstavljale tablice slučajnih brojeva dobivene preko igara na sreću. Danas, razvojem računala, one su zamijenjene generatorima slučajnih brojeva. Generiranje slučajnih brojeva je generiranje niza brojeva iz neke distribucije koji ne mogu unaprijed biti predviđeni. Različite potrebe za slučajnošću dovele su do različitih metoda generiranja slučajnih podataka. Dobar generator obuhvatit će sva važna statistička svojstva istinski slučajnog niza. Uglavnom se generiraju realizacije nizova nezavisnih uniformnih $\boldsymbol{U}(0,1)$ slučajnih varijabli.

3.1 Generatori

Dva su osnovna načina generiranja slučajnih brojeva. Jedan od njih je generiranje slučajnih brojeva na temelju prirodnih pojava. Algoritmi takvoga tipa nazivaju se **pravim generatorima slučajnih brojeva**. Oni generiraju slučajne brojeve iz nekih fizikalnih procesa kao što su radioaktivna emisija čestica ili neka druga kvantna pojava koja je stvarno nepredvidljiva i istinski slučajna situacija u prirodi. Istinski slučajni brojevi mogu se dobiti i na temelju signala i radio prijemnika podešenog da ne prima signal niti jedne radiostanice. U takvim slučajevima se iz radija čuje krčanje (tzv. bijeli šum koji je posljedica električnih stanja u atmosferi) koje je po svojoj prirodi slučajna pojava. To krčanje u obliku električnih impulsa može se poslati računalu koje ih pretvara u slučajne brojeve. Primjeri dobivanja istinski slučajnih brojeva su i bacanje standardne igraće kockice i izvlačenje brojeva u igri na sreću odnosno „Lotu". Bacanjem igraće kockice očekuje se pojavljivanje svih brojeva od 1 do 6 približno jednako puta, pri čemu se nikako ne može promatranjem prethodno dobivenih brojeva zaključiti koji će se broj pojaviti prilikom sljedećeg bacanja. Ista situacija je s Lotom, mogu se dulji vremenski period pratiti izvučeni brojevi, no bez obzira na prethodna kola, neće biti moguće predvidjeti brojeve koji će se izvući u sljedećem kolu. To su ujedno i primjeri u kojima svaki broj ima istu vjerojatnost da se pojavi ili bude izvučen.

Općenito, ovakva vrsta generatora zahtijeva vrlo skupu i osjetljivu izvedbu, dok samo generiranje niza brojeva može biti vrlo sporo. Fizikalne pojave i alati korišteni za mjerenje, doprinose tome da izlaz iz takvog generatora nije uniformno raspoređen zbog asimetričnih i sistemskih odstupanja koja se pojavljuju te je vrlo često potrebno dobivene podatke i dodatno programski obraditi. Problem je također i ponovno pokretanje simulacije nakon mijenjanja nekih parametara ili otkrivanja greške. Nije moguće ponovno pokrenuti fizikalni generator, nego bi se za ponavljanje simulacije trebao skladištiti niz već korištenih slučajnih brojeva pa ponovno pokrenuti simulacija za što bi bilo potrebno jako puno memorije. Također se može dogoditi da generatori istinski slučajnih brojeva ne prođu neke testove slučajnosti, što ne dovodi u pitanje valjanost tih testova niti slučajnost fizikalnih procesa već se to jednostavno može dogoditi primjerice zbog nedostataka hardvera koji bilježi i obrađuje slučajni izvor.

Druga metoda su računalni algoritmi koje nazivamo **generatorima pseudoslučajnih brojeva**. Oni mogu producirati duge nizove naizgled slučajnih rezultata koji su zapravo u

potpunosti određeni početnom vrijednošću. Pseudoslučajan proces je proces koji izgleda kao slučajan, no on to nije. Pseudoslučajni nizovi uglavnom posjeduju statističku slučajnost a generirani su u potpunosti deterministčkim procesima. Takav proces može se lakše izvršiti nego stvarni slučajni proces te se može ponovno koristiti kako bi se dobio potpuno isti niz vrijednosti. Generator pseudoslučajnih brojeva koristi jako malo prostora za pohranu i kodnih i internih podataka te je neusporedivo brži od pravog generatora. Naime, niti jedan pseudoslučajni generator ne može savršeno simulirati pravu slučajnost, što ne predstavlja problem sve dok on obuhvaća svojstva istinski slučajnog niza.

Do sada je poznat jako veliki broj dobrih i temeljito testiranih generatora. Generator je dobar ukoliko brzo proizvodi duge nizove brojeva i ima pouzdane, prijenosne i lako dostupne implementacije za mnoge programske jezike. Jedan od tih visokokvalitetnih generatora koji je do sada daleko najkorišteniji je **Mersenne twister**, **MT19937**, koji su izumili Makoto Matsumoto i Takuji Nishimura 1998. godine. Implementiran je u mnoga softverska rješenja te je i bazni generator pseudoslučajnih brojeva u mnogim programskim jezicima (pogledati [4]).
Postoje i kriptografski sigurni generatori slučajnih brojeva. Naime, u kriptografskim primjenama potrebna je dodatna sigurnost, stoga je nužno osigurati nepredvidivost brojeva u generiranom nizu. Kriptografski se generatori koriste za nizove pseudoslučajnih brojeva za koje je predikcija računski neizvediva. Monte Carlo uzorkovanje ne zahtijeva kriptografsku sigurnost pa se generatori slučajnih brojeva za Monte Carlo simulacije najčešće koriste običnim rekurzijama.

Glavna razlika i nedostatak generatora pseudoslučajnih brojeva i generatora istinski slučajnih brojeva je ta da se kod pseudoslučajnog generatora brojevi nakon određenog vremenskog perioda počinju periodično ponavljati, dok to kod generatora istinski slučajnih brojeva nije slučaj. Svojstva koja se očekuju od generiranog pseudoslučajnog niza su:

1. Slučajni brojevi u nizu trebali bi biti ravnomjerno raspoređeni unutar odabranog ograničenog skupa brojeva. Vjerojatnost pojavljivanja svakog broja treba biti približno jednaka. To bi trebalo vrijediti kako za cijeli generirani niz, tako i za bilo koji podniz generiranog niza.

2. Nizovi i podnizovi generiranih brojeva ne smiju biti korelirani, tj. niti jedan podniz u nizu ne smije ovisiti o nekom drugom podnizu.

3. Period generatora mora biti što je moguće veći.

Već je spomenuto na početku rada, da je Monte Carlo metodu prvi puta koristio von Neumann. On se za generiranje slučajnih brojeva služio metodom sredine kvadrata. Ta metoda je vrlo jednostavna te je pseudoslučajan niz moguće dobiti i bez korištenja računala. Uzme se n-teroznamenkasti prirodni broj, kvadrira se i potom uzme srednjih n znamenaka dekadskog zapisa tog rezultata, koji se zatim opet kvadrira te se postupak nastavlja. Postupak se ponavlja s tim da se uvijek uzima srednjih n znamenaka, a ukoliko bi znamenaka bilo manje, onda bi se dodao potreban broj nula s lijeve strane zapisa.
Početna vrijednost koja se unosi kao ulaz u generator u većini slučajeva naziva se **ključ** ili **sjeme** i označava s X_0. Trenutno stanje generatora slučajnih brojeva zadržava se u vektoru koji se naziva **vektor stanja**, pri čemu je i_0 početno stanje deterministčkog ciklusa generatora za vrijednost X_0. Duljina niza prije nego što se niz počne ponavljati označava se s **P** i naziva

periodom niza. Očito je da će male vrijednosti perioda P davati loše simulacije nasumičnog ponašanja te su zbog toga preferirane veće vrijednosti perioda. Izlaz generatora naziva se **pseudoslučajnim nizom brojeva**.

Ukoliko se postavi sjeme s recimo 4 znamenke, to jest neka je primjerice $X_0 = 6632$, tada se von Neumannovom metodom sredine kvadrata dobiva generirani niz brojeva kojemu su prva tri člana $9834, 7075, 0556, \ldots$. Naime, $6632^2 = 43\mathbf{9834}24$, $9834^2 = 96\mathbf{7075}56$, $7075^2 = 50\mathbf{0556}25$, ... Tako generirani niz nije kvalitetan u većini slučajeva jer ima jako kratak period. Može se uočiti da što je broj znamenaka sjemena manji, period će biti kraći, stoga se ovaj pristup generiranja slučajnih brojeva danas ne koristi.

Većina modernih slučajnih generatora bazirana je na jednostavnoj rekurziji koristeći modularnu aritmetiku. Najbolji primjer takvog generatora je **linearni kongruencijski generator** ili skraćeno (**LCG**) kojeg je 1948. izumio Lehmer[5]. Dva najkorištenija generatora prethodnog oblika su višestruko rekurzivni te pomaknuti Fibbonacijev generator. Zatim su tu i **nelinearni kongruencijski generatori** među kojima je najpoznatiji inverzni kongruencijski generator. Osim kongruencijskih generatora postoji još vrsta generatora koji ovdje neće biti navedeni, a više o njima i svojstvima spomenutih generatora može se pronaći u [2].

Cijeli koncept Monte-Carlo metoda temelji se na formiranju realizacija slučajnih uzoraka. Za korištenje metoda opisanih u ovom radu nužno je generirati nizove slučajnih brojeva sa željenim svojstvima. Uglavnom će u softveru koji se koristi postojati funkcija koja će to učiniti za nas. Programski jezik korišten u nastavku ovoga rada je R. On zajedno s većinom ostalih analitičkih paketa ne generira prave slučajne brojeve, već pseudoslučajne. Koristi se generatorom pseudoslučajnih brojeva zvanim $Mersenne-Twister$ čija svojstva su slična svojstvima spomenute klase generatora, ali s puno većom duljinom ciklusa. Ugrađene funkcije u $R-u$ su dobro poznate ili dobro karakterizirane distribucije poput uniformne, eksponencijalne, normalne, studentove, geometrijske, gamma i tako dalje. Međutim, moguće je generirati slučajne brojeve iz distribucija za koje neće postojati nijedna ugrađena funkcija. Metode opisane u ovome radu oslanjaju se na mogućnost generiranja beskonačnih nizova slučajnih varijabli iz neke distribucije pomoću računala. Takva se simulacija temelji na generiranju realizacija uniformne slučajne varijable na intervalu $(0, 1)$, za što se koristi odabran generator pseudoslučajnih brojeva.

Osnovni uniformni generator u R-u je funkcija **runif**() kojoj je jedini potrebni unos broj vrijednosti koje treba generirati. Ostali neobavezni parametri su min i max koji karakteriziraju granicu intervala, a zadane vrijednosti su min = 0 i max = 1. Naredbom $set.seed()$ postavlja se sjeme. Poznavanjem sjemena može se točno reproducirati cijeli niz, odnosno svaka simulacija će početi istim brojem i izlaz generatora će davati iste pseudoslučajne brojeve svaki put kada je pokrenut. To je vrlo korisno jer mogućnost ponavljanja pokusa znači da su rezultati provjerljivi. U pravilu nije nužno definirati sjeme, osim u situacijama u kojima je potrebno reproducirati potpuno isti niz slučajnih simulacija radi usporedbe. Ako generator nije inicijaliziran prije nego što počne generirati pseudoslučajne brojeve, onda ga R sam inicira koristeći vrijednost iz nekog skrivenog izvora kao što je na primjer sistemski sat.[6] Ostali dostupni generatori u R-u koji će biti korišteni u radu su: $rexp()$, $rgeom()$, $rbeta()$, $rnorm()$ i $rt()$.

[5]Derrick Henry Lehmer (1905.-1991.) američki matematičar

[6]Sistemski sat naziv je za sat koji se koristi u nekom računarskom sistemu s kojem se mjeri prolazak vremena.

3.2 Statistički testovi

Pod pojmom ispitivanja generatora misli se na ispitivanje kvalitete generiranog niza. Pri ispitivanju se želi ustanoviti pokazuje li generirani niz poželjne karakteristike uniformnosti i nezavisnosti ili ne. Nijedan univerzalan test, ili skup testova, ne može garantirati da je generator koji prođe test u potpunosti pouzdan za sve vrste simulacija. U pravilu će loši generatori pasti na vrlo jednostavnim testovima dok će oni dobri pasti samo na složenima. Ukoliko generator ne zadovoljava određene testirane karakteristike niza isti može biti odbačen kao neslučajan. Nasuprot tome, ukoliko zadovoljava odnosno prolazi sve testirane karakteristike može se s velikom vjerojatnošću tvrditi da je generator slučajan.

Ti testovi dijele se u dvije različite skupine, empirijske i teorijske testove. Empirijski testovi provode se na grupama brojeva generiranog niza za koje se procjenjuju određene statistike i ne zahtijevaju nikakvo znanje o tome kako je niz proizveden, dok teorijski testovi zahtijevaju poznavanje strukture generatora i karakteristika niza pri čemu niz ne mora nužno biti generiran. Budući se u ovome radu koristi programski jezik koji već ima ugrađene i dobro testirane generatore sa željenim svojstvima, pozornost će biti na takozvanim „*Goodness of fit* " procedurama koje su temelj empirijskih testova. Koriste se za provjeru podudarnosti simulacijskih podataka s određenim statističkim modelom. Osobito su važne za generirane nizove čiji izlaz bi trebao biti u skladu s realizacijama n.j.d. niza $\mathbf{U}(0,1)$ slučajnih varijabli. Također se koriste za testiranje rezultata asimptotskih svojstava u Monte Carlo metodama.
„*Goodness of fit* " pristup otprilike se svrstava u tri kategorije ([2], 333. *str.*):

1. Grafički prikazi poput histograma i q-q grafikona koji se smatraju neformalnim načinom usporedbe empirijske funkcije distribucije podataka s predloženom funkcijom distribucije

2. Statistički testovi temeljeni na empirijskoj funkciji distribucije

3. Statistički testovi temeljeni na kategoriziranim podacima.

Općenito se problem testiranja statističke hipoteze na temelju realizacija $(x_1, \ldots, x_n)$ slučajnog vektora $(X_1, \ldots, X_n)$ svodi na donošenje odluke o prihvaćanju ili odbacivanju hipoteze. Odluka o postupku testiranja hipoteza donosi se na temelju pravila za odbacivanje hipoteze i realizacije uzorka. Pravilo po kojem se odbacuje postavljena hipoteza podijelit će skup svih mogućih realizacija slučajnog vektora statističkog modela na dva disjunktna dijela C_r i C_r^c. Područje odbacivanja C_r postavljene hipoteze naziva se **kritično područje**. U statističkom testu testiraju se dvije hipoteze. Razlog tome je što se u hipotezu $\mathcal{H}_0$ izdvaja podskup distribucija iz statističkog modela $\mathcal{P}$, dok će hipoteza $\mathcal{H}_1$ obuhvatiti sve distribucije koje nisu sadržane u $\mathcal{H}_0$. Hipoteza $\mathcal{H}_0$ naziva se $\mathbf{nul-hipoteza}$, a $\mathcal{H}_1$ **alternativna hipoteza**. Proces donošenja odluke prihvaćanja promatrane hipoteze $\mathcal{H}_0$ definiran je onda ako je odabrana veličina uzorka n i ako je definirano kritično područje C_r. Tako definirani postupak naziva se **statističkim testom**. Odluka donesena statističkim testom može biti ispravna ili pogrešna.

Dvije su mogućnosti pogrešne odluke:

pogreška I. tipa : *odbaciti $\mathcal{H}_0$ ako je ona istinita*

pogreška II. tipa : *ne odbaciti $\mathcal{H}_0$ ako je $\mathcal{H}_1$ istinita.*

Odabir kritičnog područja temelji se na izboru maksimalne vjerojatnosti pogreške prvog tipa koja se želi prihvatiti. Odabrana maksimalna vjerojatnost pogreške prvog tipa naziva se **razina značajnosti testa** i označava s α. Uglavnom se uzimaju brojevi 0.01, 0.05 ili 0.1 za razinu značajnosti testa.
Za procjenu funkcije distribucije $F(x)$ gleda se relativna frekvencija događaja $(X_1 \leqslant x)$ u danom uzorku. Procjenitelj funkcije distribucije je

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{X_i \leqslant x\}}.$$

Tako definiran procjenitelj funkcije distribucije na temelju realizacije jednostavnog slučajnog uzorka zove se **empirijska funkcija distribucije**. Želi li se testirati dolaze li podaci, koji su realizacije jednostavnog slučajnog uzorka distribucije F, iz pretpostavljene distribucije F_0, najčešće se koriste takozvani χ^2-test i Kolmogorov-Smirnovljev test.

3.2.1 χ^2 test

Neka je dan statistički model jednostavnog slučajnog uzorka iz diskretne distribucije s konačno mnogo vrijednosti.

$$X = \begin{pmatrix} a_1 & a_2 & \dots & a_k \\ p_1 & p_2 & \dots & p_k \end{pmatrix},\ p_j > 0,\ \sum_{i=1}^{k} p_j = 1.$$

Neka je $\mathbf{p} = (p_1, \dots, p_k)$ vektorski parametar. S $\mathbf{p}_0$ je označena vrijednost parametra koja karakterizira zadanu distribuciju za hipotezu koja se želi testirati i f_j frekvencija vrijednosti a_j. Test za testiranje hipoteza

$$\mathcal{H}_0 : \mathbf{p} = \mathbf{p}_0 \tag{1}$$
$$\mathcal{H}_1 : \mathbf{p} \neq \mathbf{p}_0 \tag{2}$$

kreirat će se korištenjem generaliziranog kvocijenta vjerodostojnosti

$$\lambda(\mathbf{x}) = \frac{p_{01}^{f_1} \cdots p_{0k}^{f_k}}{\underset{\mathbf{p}}{max}\ p_1^{f_1} \cdots p_k^{f_k}}.$$

Funkcija gustoće u ovom modelu ima oblik:

$$f(\mathbf{x}; p_1, \dots, p_k) = p_1^{f_1} \cdots p_k^{f_k}.$$

ML procjenitelj za $\mathbf{p}$ dobije se maksimizacijom funkcije vjerodostojnosti po $\mathbf{p}$ uz uvjet $p_k = 1 - \sum_{j=1}^{k-1} p_j$, te izgleda: $\hat{\mathbf{p}} = (\frac{f_1}{n}, \ldots, \frac{f_k}{n})$. Dakle generalizirani kvocijent vjerodostojnosti za realizaciju $\mathbf{x}$ je:

$$\lambda(\mathbf{x}) = \prod_{i=1}^{k} \left(\frac{p_{0j}}{\hat{p}_j}\right)^{f_j},$$

odnosno

$$-2\log\lambda(\mathbf{x}) = 2\sum_{j=1}^{k-1} f_j \log\frac{\hat{p}_j}{p_{0j}}. \tag{3}$$

Obzirom da slobodnih parametara ima $k-1$, ako je H_0 istinita hipoteza, statistika generaliziranog kvocijenta vjerodostojnosti (3) po Wilksovom teoremu[7] ima asimptotsku $\chi^2(k-1)$ distribuciju. U opisu kritičnog područja testa često se koriste oznake: $f_j(emp) = n\hat{p}_j$ i $f_j(teor) = np_{0j}$. Korištenjem tih oznaka i $1-\alpha$ kvantila $\chi^2(k-1)$ distribucije, odnosno $\chi^2(k-1)_{1-\alpha}$, kritično područje ovog testa je

$$C_r = \left\{\mathbf{x} : 2\sum_{j=1}^{k-1} f_j(emp)\log\frac{f_j(emp)}{f_j(teor)} \geqslant \chi^2(k-1)_{1-\alpha}\right\}.$$

U statističkoj primjeni se za testiranje hipoteza (1) i (2) najčešće koristi takozvana Pearsonova χ^2-statistika

$$D = \sum_{j=1}^{k} \frac{[f_j(emp) - f_j(teor)]^2}{f_j(teor)}. \tag{4}$$

Statistika D iz prethodne jednakosti (4) prema Fisherovom teoremu[8] ima asimptotski $\chi^2(k-1)$ distribuciju ukoliko je $\mathbf{p}_0$ potpuno određen, a za nepoznate parametre u teorijskoj distribuciji $\mathbf{p}_0 = \mathbf{p}_0(\theta)$, pri čemu je l dimenzija od θ, $\chi^2(k-l-1)$ distribuciju. Stoga, test koji koristi tu statistiku zove se Pearsonov $\boldsymbol{\chi^2 - \text{test}}$.

Provođenje χ^2-testa zahtijeva da su podaci razvrstani u konačno mnogo razreda. Pri tome, rezultati testa u velikoj mjeri ovise o tome kako su kreirani razredi pa jedan odabir razreda može dovesti do odbacivanja hipoteze o pripadnosti zadanoj distribuciji, dok drugi odabir može rezultirati ne odbacivanjem iste hipoteze. Stoga se taj test preporučuje samo u slučaju modela koji koristi diskretnu distribuciju, pri čemu su kategorije smisleno povezane sa stvarnom varijablom o kojoj se zaključuje.

3.2.2 Kolmogorov-Smirnovljev test

Za testiranje hipoteza o jednakosti neprekidne distribucije $F_n(x)$ i pretpostavljene distribucije F_0, kreiran je cijeli niz testova temeljen na nekoj mjeri odstupanja pretpostavljene distribucije F_0 od empirijske $F_n(x)$. Osnovu za kreiranje tih testova daje takozvana vjerojatnosna integralna

[7] Wilksov teorem: Neka je $\mathcal{F}$ regularan statistički model, $\theta = (\theta_1, \ldots, \theta_k)$. Neka je $\Theta_0 \subseteq \Theta$ r-dimenzionalan skup i nul hipoteza $\mathcal{H}_\prime : \theta \in \Theta_0$. Tada statistika $-2\log\lambda(\mathbf{X})$, ako je $\mathcal{H}_\prime$ istinita, konvergira po distribuciji prema $\chi^2(k-r)$ distribuiranoj varijabli.

[8] Pogledati ([8], 241.str)

transformacija $U = F(X)$, koja komponira funkciju distribucije sa slučajnom varijablom te distribucije.

Teorem 3.1. *Neka je X neprekidna slučajna varijabla s funkcijom distribucije F za koju je dobro definiran inverz F^{-1} na intervalu $(0,1)$, tada vrijedi:*

1. *Slučajna varijabla $U = F(X)$ ima uniformnu distribuciju na $(0,1)$, odnosno $U \sim \mathbf{U}(0,1)$.*
2. *Ako je $U \sim \mathbf{U}(0,1)$, tada $F^{-1}(U)$ ima istu distribuciju kao X .*

Dokaz. Neka je $U = F(X)$. Prva tvrdnja teorema slijedi iz toga da za sve $u \in (0,1)$ vrijedi

$$\begin{aligned} F_U(u) &= P(U \leqslant u) = P(F(X) \leqslant u) \\ &= P(X \leqslant F^{-1}(u)) = F(F^{-1}(u)) = u, \end{aligned}$$

također vrijedi i

$$\begin{aligned} P(U \leqslant u) &= 0, \ za\ u \leqslant 0, \\ P(U \leqslant u) &= 1, \ za\ u \geqslant 1, \end{aligned}$$

što je upravo funkcija distribucije $\mathbf{U}(0,1)$.

Druga tvrdnja vrijedi zbog činjenice da je $\forall x \in \mathbb{R}$

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

□

Kolmogorov-Smirnovljev test temelji se na funkciji

$$d(\mathbf{x}) = \sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F_0(x)|,$$

gdje su $\mathbf{x} = (x_1, \ldots, x_n)$ realizacije slučajnog vektora $\mathbf{X} = (X_1, \ldots, X_n)$ i sljedećem teoremu.

Teorem 3.2. *U modelu jednostavnog slučajnog uzorka $\mathbf{X} = (X_1, \ldots, X_n)$ iz distribucije F koja zadovoljava uvjete teorema* 3.1*, pod pretpostavkom istinitosti hipoteze*

$$\mathcal{H}_0 : F = F_0$$

statistika $D_n = d(\mathbf{X})$ ima istu distribuciju neovisno o F_0. Specijalno,

$$P_{F_0}(D_n \leqslant d) = P_U(D_n \leqslant d), \ \mathbf{U}(0,1).$$

Dokaz. Neka je $X = (X_1, \ldots, X_n)$ jednostavan slučajan uzorak iz neprekidne distribucije F, neka je istinita nulta hipoteza $\mathcal{H}_0 : F = F_0$ i neka vrijedi $U_i = F_0(X_i)$. Zbog istinitosti hipoteze $\mathcal{H}_0$ su $U_i \sim \mathbf{U}(0,1)$ i međusobno nezavisne $\forall i = 1, \ldots, n$. Vrijedi sljedeće:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,x]}X_i = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,F_0(x)]}F_0(X_i) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,F_0(x)]}(U_i)$$

Dakle, $F_n(x) = G_n(F_0(x))$, gdje je G_n empirijska funkcija distribucije od $(U_1, \ldots, U_n)$. Također vrijedi:

$$\sup_{x\in\mathbb{R}} |F_n(x) - F_0(x)| = \sup_{x\in\mathbb{R}} |G_n(F_0(x)) - F_0(x)| = \sup_{u\in(0,1)} |G_n(u) - u|.$$

□

Za konstrukciju testa potrebno je znati distribuciju od D_n ako je H_0 istinita. Prethodni teorem omogućuje da se njena distribucija utvrdi pomoću simulacija iz $\mathbf{U}(0,1)$.
Dakle, ako je $X = (X_1, \ldots, X_n)$ jednostavan slučajan uzorak iz neprekidne distribucije F, za testiranje hipoteza

$$\mathcal{H}_0 : F = F_0$$
$$\mathcal{H}_1 : F \neq F_0$$

koristi se test statistika

$$D_n = \sup_{x\in\mathbb{R}} |\hat{F}_n(x) - F_0(x)|,$$

pri čemu velike vrijednosti ove statistike idu u prilog alternativnoj hipotezi. Test koji koristi tu statistiku D_n naziva se **Kolmogorov − Smirnovljev** test. Distribucija ovisi o n te je njeno korištenje omogućeno u standardnim statističkim paketima.

Problem KS-testa je u njegovoj relativno lošoj razlučivosti. To znači da, ukoliko rezultat KS-testa upućuje na odbacivanje H_0, tada je prilično sigurno da podaci ne potječu iz F_0. Međutim, u slučaju ne odbacivanja H_0 može se dogoditi da se ipak radi o podacima iz neke druge distribucije. Loša razlučivost KS-testa motivira kreiranje boljih testova u tom smislu, specijalno za pojedine distribucije koje se često javljaju u primjeni. Alternativa KS testu je Anderson-Darling test čiji se detaljan opis može pronaći u [2].

4 Neuniformne slučajne varijable

Uniformne slučajne varijable osnovni su sastavni dio Monte Carlo metoda. Često je prva stvar koja se čini s uniformnim slučajnim varijablama konstruiranje neuniformne slučajne varijable s obzirom na zadani problem. Najraširenije matematičko računalno okruženje uključuje generatore s velikim izborom neuniformnih distribucija poput normalne, Poissonove, binomne, eksponencijalne, gamma, i tako dalje. Ponekad se uzima i uzorak iz distribucije koju računalno okruženje ne uključuje, stoga je potrebno znati generirati realizacije neuniformne slučajne varijable. U ovom će se poglavlju opisati metode za transformaciju uzoraka iz uniformne distribucije u uzorke iz drugih distribucija, metodom inverzije i metodom odbijanja, što je vrlo bitno jer se na tom principu baziraju i metode za generiranje slučajnih vektora i procesa kao i princip na koji djeluje Markov Chain Monte Carlo [2].

4.1 Metoda inverzije

Najizravniji način pretvaranja uniformne varijable u neuniformnu slučajnu varijablu je invertiranjem funkcije distribucije. U pravilu ju je moguće izvesti za svaku funkciju distribucije i primjenjiva je za vrlo jake tehnike redukcije varijance o čemu će biti govora u poglavlju (5). Sam postupak invertiranja nije uvijek tako jednostavan pa će biti razmotrene i alternative.

Metoda generira slučajne brojeve iz bilo koje funkcije gustoće koristeći njezinu inverznu funkciju distribucije $F^{-1}(x)$, odnosno generalizirani inverz funkcije distribucije.

Definicija 4.1. *Neka je X slučajna varijabla s funkcijom distribucije F, definiramo generalizirani inverz funkcije distribucije F^{-1} na $(0,1)$ s*

$$F^{-1}(u) = \inf\{x : F(x) \geqslant u\},\ 0 < u < 1. \tag{5}$$

Kao posljedica teorema (3.1), za generiranje n podataka iz neprekidne slučajne varijable s distribucijom F, dovoljno je imati niz podataka $(u_i,\ i = 1, \ldots, n)$ iz $\mathbf{U}(0,1)$. Traženi se podaci tada dobiju kao $(F^{-1}(u_i),\ i = 1, \ldots, n)$.

Teorem 4.2. *Ako je $U \sim \mathbf{U}(0,1)$ tada je i $1-U \sim \mathbf{U}(0,1)$. Također vrijedi i $F^{-1}(1-U) \sim F$.*

Dokaz. Neka je $U \sim \mathbf{U}(0,1)$, treba pokazati da funkcija distribucije $F_{1-U}(x)$ odgovara uniformnoj $\mathbf{U}(0,1)$. Neka je $x \in [0,1]$, tada je

$$\begin{aligned} F_{1-U}(x) &= P(1-U \leqslant x) = P(-U \leqslant x-1) = P(U \geqslant 1-x) = 1 - P(U \leqslant 1-x) \\ &= 1 - F_U(1-x) = 1 - (1-x) = x. \end{aligned}$$

Treba se još pokazati da je $F_{1-U}(x) = 0$, za $x < 0$ i $F_{1-U}(x) = 1$, za $x > 1$.

Za $x < 0$ je

$$F_{1-U}(x) = P(1-U \leqslant x) = P(U \geqslant 1-x) = 1 - P(U \leqslant 1-x) = 1 - F_U(1-x)$$

kako je $1 - x > 1$ imamo da je $F_{1-U}(x) = 0$.

Analogno, za $x > 1$ je $F_{1-U}(x) = 1$.

$$\Rightarrow F_{1-U}(x) = \begin{cases} 0,\ x < 0 \\ x,\ x \in [0,1] \\ 1,\ x > 1 \end{cases} \sim \mathbf{U}(0,1).$$

Drugi dio teorema direktna je posljedica 2. tvrdnje iz teorema (3.1).

□

Do sada je definirana inverzija za rad s uniformnim slučajnim varijablama $\mathbf{U}(0,1)$, no ideja djeluje općenitije.

Ako je F neprekidna funkcija distribucije, G bilo koja funkcija distribucije i $F(X) \sim \mathbf{U}(0,1)$, tada iz

$$Y = G^{-1}(F(X)), \tag{6}$$

slijedi da će Y imati distribuciju G ([7]).
Da bi se to dokazalo, uočimo da je $F(X) \sim \mathbf{U}(0,1)$ i $Y = G^{-1}(F(X))$ pa slijedi da će distribucija $F_Y(y)$ biti jednaka

$$F_Y(y) = P(G^{-1}(F(X)) \leqslant y) = P(F(X) \leqslant G(y)) = F_{F(X)}(G(y)).$$

Kako je $F(X) \sim \mathbf{U}(0,1)$, vrijedi

$$F_Y(y) = F(G(y)) = G(y).$$

Funkcija $G^{-1}(F(\cdot))$ transformira kvantile distribucije F u odgovarajuće kvantile od G. Budući da je funkcija kvantila[9] upravo generalizirani inverz funkcije distribucije, $G^{-1}(F(\cdot))$ naziva se QQ transformacija (kvantil-kvantil transformacija). Ponekad $G^{-1}(F(\cdot))$ ima jednostavniji oblik i od G^{-1} i od F pa je moguće direktno iz $X \sim F$ doći do $Y = (G^{-1} \circ F)(X) \sim G$ bez prolaska kroz $\mathbf{U}(0,1)$. Primjer takve konstrukcije je $Y = \mu + \sigma Z$, $\sigma > 0$, za generiranje realizacija iz $Y \sim \mathcal{N}(\mu, \sigma^2)$ preko $Z \sim \mathcal{N}(0,1)$. Naime,

$$P(Y \leqslant y) = P(\mu + \sigma Z \leqslant y) = P\Big(Z \leqslant \frac{y-\mu}{\sigma}\Big) = F_Z\Big(\frac{y-\mu}{\sigma}\Big),$$

gdje se deriviranjem dobije

$$f\Big(\frac{y-\mu}{\sigma}\Big) \cdot \frac{1}{\sigma} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}.$$

Slijedi da $Y = \mu + \sigma Z$ ima $\mathcal{N}(\mu, \sigma^2)$ distribuciju.

Algoritam 4.3. *(Inverzna transformacija za neprekidnu distribuciju)*
Uz pretpostavku da se žele generirati realizacije slučajne varijable X s funkcijom distribucije $F_X(x)$, algoritam inverzne transformacije za neprekidnu distribuciju je sljedeći:

1. *Generiraj u iz $\mathbf{U}(0,1)$*
2. *Vrati $x = F_X^{-1}(u)$.*

Zatim će niz tako generiranih brojeva slijediti distribuciju $F_X(x)$.

Treba imati na umu da ovaj algoritam radi općenito, ali nije uvijek praktičan. Na primjer, pretvaranje F_X je lako ako je X eksponencijalna slučajna varijabla, ali znatno teže ako je X normalna slučajna varijabla.

[9] Funkcija kvantila za neku slučajnu varijablu X s funkcijom distribucije $F(x)$ definira se kao $Q(p) = \inf\{x : F(x) \geqslant p\},\ 0 < p < 1$.

Primjer 4.4. *Simuliranje eksponencijalne slučajne varijable korištenjem metode inverza.*

Ako je $X \sim Exp(\lambda)$*, tada* X *ima funkciju gustoće* $f(x) = \lambda e^{-\lambda x}$ *za* $x > 0$ *i funkciju distribucije* $F_X(x) = P(X \leqslant x) = \begin{cases} 1 - e^{-\lambda x}, & x > 0 \\ 0 \quad , & inače \end{cases}$

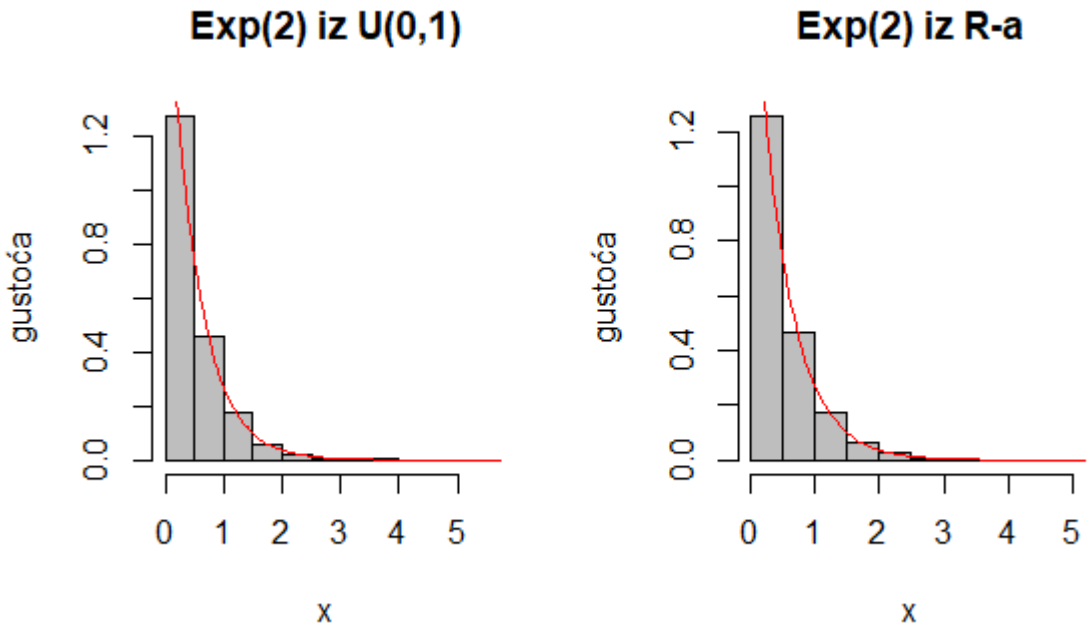
Funkcija inverza dana je s

$$F_X^{-1}(x) = -\frac{\ln(1-x)}{\lambda}.$$

Pomoću gornjeg algoritma, prvo se generiraju realizacije u *iz* $\mathbf{U}(0,1)$*, zatim postavlja* $x = F_X^{-1}(u)$*, pa slijedi da je*

$$X \stackrel{d}{=} -\frac{\ln(1-U)}{\lambda} \stackrel{d}{=} -\frac{\ln(U)}{\lambda} \sim Exp(\lambda).$$

Grafički su u $R-u$ *prikazani histogrami slučajne varijable dobivene inverznom transformacijom za* $\lambda = 2$*, odnosno* $X = -\frac{\ln(U)}{2} \sim Exp(2)$ *i ugrađene funkcije rexp(), koja generira podatke iz eksponencijalne distribucije.*



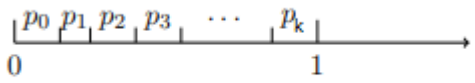
Slika 3: Histogrami eksponencijalnih slučajnih varijabli pomoću inverzne transformacije (lijevo) i R naredbom $rexp()$ (desno)

Generirana slučajna varijabla (lijevi graf) slijedi planiranu distribuciju i histogrami se gotovo ne razlikuju. Kolmogorov-Smirnovljevim testom testirana je nul hipoteza o jednakosti distribucije

dva uzorka te dobivena p-vrijednost 0.4784*, sugerira kako nema razloga odbaciti nul hipotezu o jednakosti distribucije dva generirana uzorka.*

Ako je $X = F^{-1}(U)$ za $U \sim \mathbf{U}(0,1)$, pri čemu je F funkcija distribucije diskretne slučajne varijable i F^{-1} generalizirani inverz, tada $F(X)$ neće nužno imati uniformnu $\mathbf{U}(0,1)$ distribuciju. Jednakost (6) generalno neće vrijediti za funkcije distribucije diskretne slučajne varijable. Ipak, uniformna distribucija na intervalu $(0,1)$ može se iskoristiti za generiranje realizacija diskretnih slučajnih varijabli koje uzimaju konačno mnogo vrijednosti. Pretpostavlja se da su te vrijednosti $1, \ldots, N$ te da je X diskretna slučajna varijabla takva da je $P(X = k) = p_k \geqslant 0$ i $\sum_k p_k = 1$, za $k = 1, \ldots, N < \infty$.

Prvo se generira realizacija u slučajne varijable koja ima $\mathbf{U}(0,1)$ distribuciju. Zatim se interval $[0,1]$ podijeli u podintervale tako da $k-ti$ podinterval ima duljinu p_k kao na slici (4):



Slika 4: Generiranje diskretnih slučajnih varijabli

Definiramo

$$\mathrm{X} = \begin{cases} x_0,\ u < p_0 \\ x_1,\ p_0 \leqslant u < p_0 + p_1 \\ \vdots \\ x_k,\ \sum_{i=1}^{k-1} p_i \leqslant u < \sum_{i=1}^{k} p_i \\ \vdots \end{cases}$$

odnosno $X = x_k$ ako $F(x_{k-1}) \leqslant u < F(x_k)$ gdje je $F(x)$ željena funkcija distribucije. Dobiva se sljedeće

$$P(X = x_k) = P\Big(\sum_{i=1}^{k-1} p_i \leqslant u < \sum_{i=1}^{k} p_i\Big) = p_k,$$

pa X ima traženu distribuciju.

Prethodno opisani postupak može se prikazati sljedećim algoritmom.

Algoritam 4.5. *(Inverzna transformacija za diskretnu distribuciju)*
Algoritam inverzne transformacije za diskretnu distribuciju je sljedeći:

1. *Generiraj u iz $\mathbf{U}(0,1)$*
2. *Odredi indeks k takav da je $\sum_{i=1}^{k-1} p_i \leqslant u < \sum_{i=1}^{k} p_i$ i vrati $x = x_k$.*

Primjer 4.6. *Simuliranje podataka iz diskretne slučajne varijable X koja slijedi sljedeću distribuciju:*

$$X = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0.35 & 0.15 & 0.4 & 0.1 \end{pmatrix}.$$

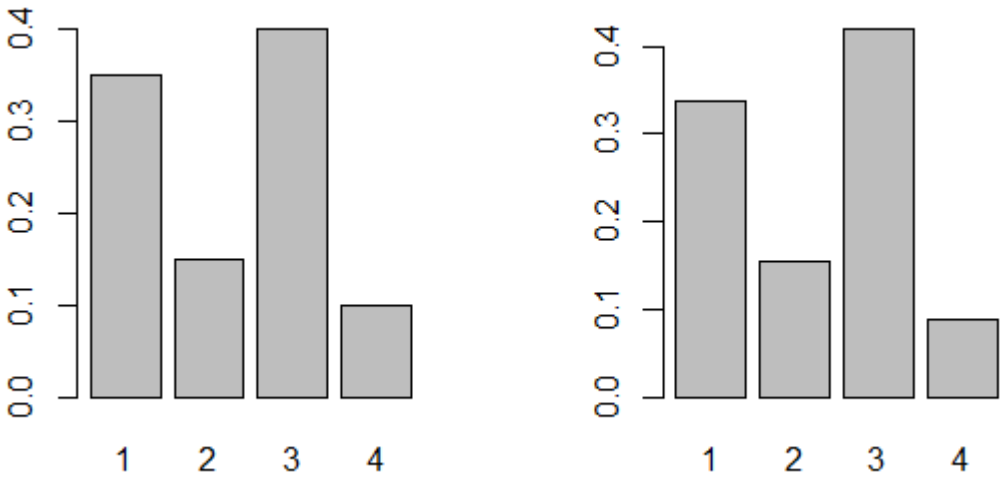
Podijeli se interval $[0,1]$ *u podintervale* A_i*, tako da svaki ima duljinu* p_i*, odnosno*

$$A_1 = [0, 0.35),\ A_2 = [0.35, 0.5),\ A_3 = [0.5, 0.9),\ A_4 = [0.9, 1).$$

Računalom se dobije broj u *te ukoliko je on iz* A_i*, onda je* $x = x_i$*, tj.* $P(X = x_i) = P(U \in A_i) = p_i$*.*

Prvim pokretanjem koda dobije se $u = 0.170642$ *koji je iz intervala* $A_1 = [0, 0.35)$ *pa slijedi da je* $x = 1$*. U* 1000 *simulacija dobivena je sljedeća funkciju distribucije :*

$$X = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0.337 & 0.155 & 0.419 & 0.089 \end{pmatrix}.$$



Slika 5: Stupčasti dijagrami podataka zadane funkcije distribucije (lijevo) i empirijske funkcije distribucije (desno)

Iz grafičkog prikaza vidljivo je da generirane realizacije slučajne varijable (desni graf) slijede planiranu distribuciju. Za testiranje hipoteze o jednakosti diskretnih distribucija koje nemaju velike skupove stanja moguće je koristiti χ^2 *- test. Testirat će se nul-hipoteza* $\mathbf{p} = \mathbf{p}_0$*, odnosno da uzorak potječe iz distribucije s pripadnim vjerojatnostima* $\mathbf{p}$*, kao što je opisano u poglavlju* (3.2)*. Budući je p-vrijednost jednaka iznosu* 0.2133 *može se reći kako podaci ne podupiru tvrdnju o odbacivanju nul hipoteze.*

Primjer 4.7. *Simuliranje realizacija geometrijske slučajne varijable koristeći metodu inverza.*

Funkcija gustoće s parametrom $p \in (0,1)$ *je* $p_k = P(X = k) = p(1-p)^{k-1}$, *za* $k = 1, 2, 3, \ldots,$

dok je funkcija distribucije

$$\begin{aligned} F_X(x) = P(X \leqslant x) &= 1 - P(X > x) = 1 - \sum_{k=x+1}^{\infty} p_k = 1 - \sum_{k=x+1}^{\infty} p(1-p)^{k-1} \\ &= 1 - p(1-p)^x[1 + (1-p) + (1-p)^2 + \ldots] = 1 - p(1-p)^x\Big[\frac{1}{1-(1-p)}\Big] \\ &= 1 - (1-p)^x, \quad za\ x = 1, 2, \ldots. \end{aligned}$$

Za $0 < u < 1$ *trebamo pronaći takav* $k \in \mathbb{N}$ *za koji je* $1 - (1-p)^{k-1} < u \leqslant 1 - (1-p)^k$

Dobivamo

$$k - 1 < \frac{log(1-u)}{log(1-p)} \leqslant k,$$

stoga je

$$k = \left\lfloor \frac{log(1-u)}{log(1-p)} \right\rfloor. \tag{7}$$

Ponekad se geometrijska distribucija definira kao broj pokušaja prije prvog uspjeha, uključujući i taj uspjeh, odnosno $p_k = P(X = k) = p(1-p)^k$ *za* $k = 0, 1, 2, 3, \ldots$ *gdje se tada* $\lfloor \cdot \rfloor$ *u jednakosti* (7) *mijenja s* $\lceil \cdot \rceil$.

Općenito metoda inverzne transformacije za diskretne slučajne varijable generalno ne predstavlja neki problem, no za neprekidne distribucije integriranje funkcije gustoće analitički je nemoguće za većinu distribucija, uključujući i normalnu. U nekim se situacijama nema izravna inverzna distribucija F^{-1}, ali se može invertirati numerički. U najboljem slučaju $F(x)$ je dostupna eksplicitno. Ukoliko je F neprekidna, x se može tražiti numerički. Ako F nije zadana eksplicitno, ali njena funkcija gustoće $f(x) = F'(x)$ je, tada se aproksimira funkcija distribucije F numeričkom integracijom i potom se invertira (pogledati [7]). Samim time, ovo nije najefikasnija metoda generiranja realizacija slučajnih varijabli. Metoda koja je alternativa metodi inverzne transformacije za generiranje realizacija diskretnih slučajnih varijabli je *Alias metoda*. Ona ne zahtijeva dugotrajno pretraživanje kao u 2. koraku algoritma (4.5). Više o toj metodi može se pronaći u [2].

4.2 Metoda odbacivanja

Za razliku od metode inverzije koja se direktno bavi funkcijom distribucije slučajne varijable čije se realizacije žele generirati, metoda odbacivanja može se smatrati indirektnom metodom. Indirektnom u smislu da se generiraju realizacije slučajne varijable i prihvaćaju samo kako bi se

mogle podvrgnuti testu kojim se donosi konačna odluka o prihvaćanju te realizacije. Uveli su je Ulam i von Neumann, naziva se još i metoda prihvaćanja-odbijanja, a primjenjiva je za svaku funkciju gustoće iz $\mathbb{R}^n$. Primjenjuje se u slučaju kada prethodno spomenuta direktna metoda zakaže ili se pokaže računski neučinkovitom. Metoda simulira slučajan uzorak iz nepoznate, komplicirane distribucije koja se još naziva i ciljana distribucija F_X s funkcijom gustoće $f(x)$, koristeći slučajne uzorke iz poznate ili predložene distribucije F_Y, koja se još naziva kandidat distribucija, s funkcijom gustoće $g(x)$. Metoda odbacivanja „odbija" neke generirane realizacije, a neke prihvaća. Cilj je da prihvaćene realizacije budu distribuirane kao da su iz ciljane distribucije. Da bi se uopće mogla koristiti metoda odbacivanja mora vrijediti da je nosač funkcije f podskup nosača funkcije g.
Ako je χ_f nosač funkcije f i χ_g nosač od g, mora vrijediti $\chi_f \subset \chi_g$. Kada bi postojalo područje nosača funkcije f koje g nikad ne bi diralo tada taj dio nikad ne bi bio uzorkovan. Pored toga, mora se pretpostaviti da je

$$c = \sup_{x \in \chi_f} \frac{f(x)}{g(x)} < \infty$$

i da se c može izračunati. Najlakši način da se ta pretpostavka zadovolji je da se osigura da funkcija g ima teže repove od f, u suprotnom metoda neće raditi.

Algoritam 4.8. *(Metoda odbacivanja)* ([3], 52.*str*)

1. *Generiraj u iz $\mathbf{U}(0,1)$*
2. *Generiraj y iz predložene funkcije gustoće g.*
3. *Ako je*

$$u \leqslant \frac{f(y)}{cg(y)}$$

tada prihvati y, tj. vrati $x = y$, u suprotnom ga odbaci i vrati se na 1. korak.

Algoritam se može ponavljati dok željeni broj uzoraka iz ciljane funkcije gustoće f ne bude prihvaćen.
Teorijska podloga ove metode dana je sljedećom propozicijom.

Propozicija 4.9. ([3], 52.*str.*)
Slučajna varijabla X čije su realizacije generirane metodom odbacivanja ima funkciju gustoće f.

Dokaz. Teorem će se dokazati korištenjem uvjetnih i marginalnih distribucija slučajnog vektora (Y, U). Neka je $f(y, u)$ oznaka za funkciju gustoće slučajnog vektora (Y, U). Naime, budući da su Y i U nezavisne, tada je

$$f(y,u) = f(y) \cdot f(u) = f(y) \cdot 1 = g(y), \ za\ u \in (0,1),$$

u suprotnom je $f(y, u) = 0$. Dakle, može se označiti $f(y, u) = g(y)$ za $u \in (0, 1)$.
Vrijedi sljedeće

$$
\begin{aligned}
F_{Y|U}(x) &= P(Y \leqslant x \mid U \leqslant f(Y)/cg(Y)) \\
&= \frac{P(Y \leqslant x \ , \ U \leqslant f(Y)/cg(Y))}{P(U \leqslant f(Y)/cg(Y))} \\
&= \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/cg(y)} g(y) \ du \ dy}{\int_{0}^{f(y)/cg(y)} \int_{-\infty}^{\infty} g(y) \ du \ dy} \\
&= \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/cg(y)} g(y) \ du \ dy}{\int_{-\infty}^{\infty} g(y) \ dy \int_{0}^{f(y)/cg(y)} \ du} \\
&= \frac{\int_{-\infty}^{x} g(y) \ [f(y)/cg(y)] \ dy}{\int_{-\infty}^{\infty} g(y) \ [f(y)/cg(y)] \ dy} \\
&= \frac{\int_{-\infty}^{x} f(y) \ dy}{\int_{-\infty}^{\infty} f(y) \ dy}
\end{aligned}
$$

$$
\Rightarrow P(X \leqslant x) = \int_{-\infty}^{x} f(y) \ dy.
$$

□

Napomena 4.10. *Svojstva metode odbijanja:*

1. *Neka je I slučajna varijabla koja predstavlja broj iteracija metode odbijanja, odnosno*

$$
I = min\left\{ k \geqslant 1 : U_k < \frac{f(Y_k)}{cg(Y_k)} \right\}.
$$

Tada $\forall \ j \geqslant 1$ i $p \in (0,1)$ vrijedi

$$
\begin{aligned}
P(I = j) &= P\left(U_1 \geqslant \frac{f(Y_1)}{cg(Y_1)}, \ldots, U_{j-1} \geqslant \frac{f(Y_{j-1})}{cg(Y_{j-1})}, U_j < \frac{f(Y_j)}{cg(Y_j)} \right) \\
&= P\left(U_1 \geqslant \frac{f(Y_1)}{cg(Y_1)} \right) \ldots P\left(U_{j-1} \geqslant \frac{f(Y_{j-1})}{cg(Y_{j-1})} \right) P\left(U_j < \frac{f(Y_j)}{cg(Y_j)} \right) \\
&= (1-p)^{j-1} p,
\end{aligned}
$$

iz čeg slijedi da je $I \sim \mathcal{G}eo(p)$.

Budući da će svaka iteracija ove metode nezavisno rezultirati prihvaćanje vrijednosti s vjerojatnošću

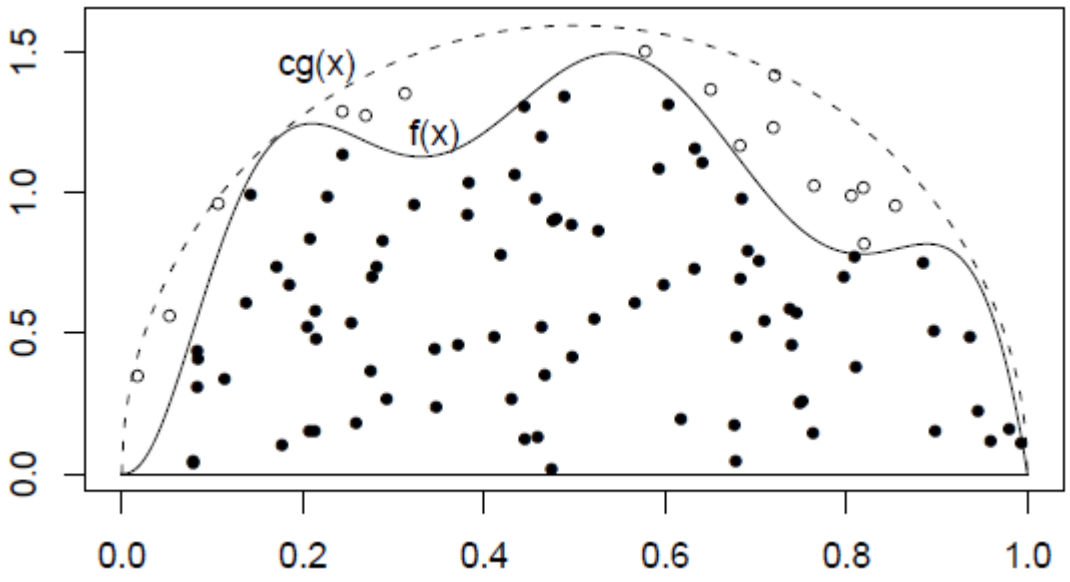
$$
\begin{aligned}
p = P(U \leqslant f(Y)/cg(Y)) &= \int_{0}^{f(y)/cg(y)} \int_{-\infty}^{\infty} g(y) \ du \ dy \\
&= \int_{-\infty}^{\infty} [f(y)/cg(y)] \ g(y) \ dy \\
&= \frac{1}{c} \int_{-\infty}^{\infty} f(y) \ dy = \frac{1}{c},
\end{aligned}
$$

slijedi da će broj iteracija imati geometrijsku distribuciju s parametrom $1/c$*, odnosno* $I \sim \mathcal{G}eo(\frac{1}{c})$*. Stoga je očekivani broj iteracija jednak* c*. Učinkovitost metode odbijanja definira se kao vjerojatnost prihvaćanja generirane vrijednosti.*

2. *Algoritam će funkcionirati za bilo koji broj* $c' \geqslant c$*, ali će biti manje učinkovit. Naime, što je funkcija gustoće* g *bliža funkciji* f*, vrijednost konstante* c *bit će manja što će po prethodnom svojstvu rezultirati većom vjerojatnošću prihvaćanja generirane realizacije.*

Teoretski, svaka funkcija gustoće g može biti izabrana za kandidata sve dok njezin nosač sadrži nosača funkcije f. Međutim, u praksi se g bira tako da se podudara s f što je više moguće. U pravilu, kandidati za g koji se usko podudaraju s f imat će manje vrijednosti konstante c i samim time će biti prihvaćeni s većom vjerojatnošću. Velike vrijednosti konstante c rezultirat će algoritmom koji odbija mnoge realizacije kandidat funkcije te će se smanjiti učinkovitost metode. Prikažimo to geometrijski.

Neka je $f(x)$ funkcija gustoće neprekidne slučajne varijable i neka je A područje ispod grafa funkcije $f(x)$, odnosno $A = \{(x, z) : 0 \leqslant z \leqslant f(x)\}$. Ako se uzorkuju točke (x, z) koje su realizacije slučajnog vektora (X, Z) s uniformnom distribucijom na A, onda je $f(x)$ marginalna funkcija gustoće od X.
Da bi se dobile takve točke (x, z) koje su realizacije uniformnog slučajnog vektora (X, Z) treba se uzorkovati područje ispod grafa funkcije $cg(x)$ i zadržavati samo one točke koje su ispod $f(x)$ kao što je prikazano na slici (6).



Slika 6: Geometrijska interpretacija metode odbijanja

Definira se skup

$$S_M(h) = \{(x, z) : 0 \leqslant z \leqslant Mh(x), x \in \mathbb{R}\} \subset \mathbb{R}^2,$$

gdje je h funkcija gustoće na $\mathbb{R}$ kao što su f ili g i M pozitivna konstanta $M > 0$ koja je po potrebi 1 ili c.

Geometrijska interpretacija metode odbijanja je iskazana sljedećim teoremom.

Teorem 4.11. ([7])
Neka je h funkcija gustoće na $\mathbb{R}$. Ako je slučajni vektor (X, Z) uniformno distribuiran na $S_M(h)$ za $M > 0$, tada je $X \sim h$.

Dokaz. Skup $S_M(h)$ će se označiti sa $S = S_M(h)$ i pretpostavit će se da je $(X, Z) \sim \mathbf{U}(S)$ te uzeti $x \in \mathbb{R}$. Tada je vjerojatnost da je točka (x, z) ispod grafa funkcije $Mh(x)$ jednaka

$$\begin{aligned} P(X \leqslant x) = P((-\infty, x] \times [0, \infty) \mid S) &= \frac{P((-\infty, x] \times [x, \infty) \cap S)}{P(S)} \\ &= \frac{\int_{-\infty}^{x} \int_{0}^{Mh(y)} 1 \, dz \, dy}{\int_{-\infty}^{\infty} \int_{0}^{Mh(y)} 1 \, dz \, dy} = \frac{\int_{-\infty}^{x} Mh(y) \, dy}{\int_{-\infty}^{\infty} Mh(y) \, dy} = \int_{-\infty}^{x} h(y) \, dy. \end{aligned}$$

□

Primjer 4.12. *Korištenje metode odbacivanja za generiranje slučajne varijable $X \sim \mathcal{B}e(2, 4)$ s funkcijom gustoće $f(x) = 20x(1 - x)^3$, za $0 < x < 1$.*

Neka je $g(x) = 1$, $0 < x < 1$.

Da bi se odredila konstanta c potrebno je izračunati maksimum vrijednosti

$$\frac{f(x)}{g(x)} = 20x(1 - x)^3.$$

Deriviranjem se dobije

$$\Big(\frac{f(x)}{g(x)}\Big)' = 20[(1 - x)^3 - 3x(1 - x)^2] = 20(1 - x)^2(1 - 4x).$$

Izjednačavanjem s 0, dobiju se nultočke $x_{1,2} = 1$ i $x_3 = \frac{1}{4}$. Kako je $0 < x < 1$ maksimalna se vrijednost postiže u $x = \frac{1}{4}$,

$$\frac{f(x)}{g(x)} \leqslant 20\Big(\frac{1}{4}\Big)\Big(\frac{3}{4}\Big)^3 = \frac{135}{64} \equiv c.$$

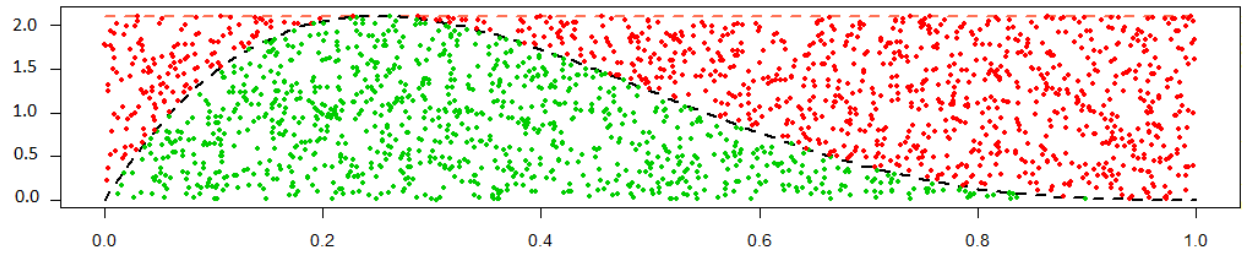
Dakle,

$$\frac{f(x)}{cg(x)} = \frac{256}{27}x(1-x)^3.$$

Postupak odbijanja je sljedeći:

1. *Generiraj* u *iz* $\mathbf{U}(0,1)$ *i* y *iz* $\mathbf{U}(0,1)$

2. *Ako je* $u \leqslant \frac{256}{27}y(1-y)^3$, *vrati* $x = y$, *u suprotnom se vrati na 1. korak.*



Slika 7: Generiranje slučajne varijable $X \sim \mathcal{B}e(2,4)$ metodom odbacivanja

Slika prikazuje rezultat generiranja 1000 *parova* (y,u) *iz* g *i* $\mathbf{U}_{[0,c]}$, *odnosno* $\mathbf{U}(0,1)\times\mathbf{U}(0,c)$, *gdje je* $c = 2.109$. *Zelene točke koje se nalaze ispod funkcije gustoće* f *(crna boja) su one vrijednosti za koje se prihvaća* $y = x$, *a odbacuju vrijednosti obojane crveno koje se nalaze izvan. Broj iteracija potrebnih za simuliranje uzorka duljine* 1000 *je* 2165, *a očekivana vrijednost iznosi* $1000c = \frac{3375}{16} = 2109.375$. *Prihvaćeno je* 47% *simuliranih vrijednosti.*

Metoda odbacivanja vrlo je moćna tehnika, ali je neučinkovita. Neučinkovita u smislu da će konačne realizacije sadržavati samo neke od svih generiranih točaka. Drugim riječima, traži se od algoritma da izvede neki dodatni posao čiji rezultati neće završiti u konačnom uzorku.

5 Monte Carlo

Monte Carlo simulacijama se procjenjuju razni matematički izrazi poput integrala, rješenja sustava jednadžbi ili kompliciranijih matematičkih modela. Većina matematičkih izraza može se procijeniti metodama numeričke analize, no Monte Carlo metode nude alternativni pristup koji je ponekad jedini prikladan za lako korištenje i kontroliranje. Čest slučaj korištenja Monte Carlo simulacija je procjena vrijednosti višedimenzionalnih određenih integrala što se može primijeniti primjerice u određivanju površine ili volumena.

U Monte Carlo problemu se određuje iznos koji se želi znati kao očekivana vrijednost slučajne varijable Y, tj. $\mu = \mathbb{E}(Y)$. Tada se nezavisno i slučajno generiraju vrijednosti $y_1, \ldots, y_n$ iz distribucije od Y i uzimaju njihove prosječne vrijednosti kao procjena od μ. Sljedeći rezultati mogu se pronaći u ([2], 305. *str.*).
Monte Carlo procjenitelj za μ dan je s

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i. \tag{8}$$

Pod uvjetom da je varijanca $\sigma^2 = Var\ Y$ konačna, zbog centralnog graničnog teorema

$$\frac{\bar{Y} - \mu}{\sigma}\sqrt{n} \ \xrightarrow{\mathrm{D}}\ \mathcal{N}(0,1),\ n \to \infty$$

slijedi da je približna distribucija Monte Carlo procjenitelja asimptotski normalna s očekivanjem μ i varijancom $\frac{\sigma^2}{n}$:

$$\bar{Y} \sim AN\Big(\mu, \frac{\sigma^2}{n}\Big), n \to \infty. \tag{9}$$

U praksi je parametar σ^2 najčešće nepoznat, ali se može procijeniti uzoračkom varijancom

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2,$$

koja po zakonu velikih brojeva ide u σ^2 kada $n \to \infty$.
Specijalno, za $\alpha \in (0,1)$ i veliki n, interval pouzdanosti $(1-\alpha)$ za očekivanje μ je:

$$\Big[\bar{Y} - z_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}},\ \bar{Y} + z_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}\Big], \tag{10}$$

gdje je $z_{1-\frac{\alpha}{2}}$, $(1-\frac{\alpha}{2})$ - kvantil standardne normalne distribucije $N(0,1)$.

Osnovni postupak procjene nezavisnih i jednako distribuiranih podataka sažet je u nastavku. Postupak se ponekad naziva i jednostavnim Monte Carlom (eng. crude Monte Carlo(CMC)).

Algoritam 5.1. *(Jednostavni Monte Carlo)* ([2], 306.*str*)

1. *Generiraj n.j.d. niz $y_1, \ldots, y_n$ iz distribucije od Y*
2. *Vrati vrijednost procjene $\bar{y}$ i interval pouzdanosti* (10) *za $\mu = \mathbb{E}(Y)$.*

Često je Y funkcija nekog slučajnog vektora ili čak i slučajnog procesa, odnosno $Y = f(\mathbf{X})$, gdje je $\mathbf{X} \in \mathcal{D} \subseteq \mathbb{R}^d$ slučajni vektor s funkcijom gustoće $p(\mathbf{x})$ i f realna funkcija definirana na $\mathcal{D}$. Također, $\mathbf{X}$ ne mora uopće biti točka u Euklidskom prostoru, on može biti i put neke lutajuće čestice, dokle god je $Y = f(\mathbf{X})$ iznos koji ima aritmetičku sredinu, kao što ju imaju realni brojevi ili vektori, može se koristiti Monte Carlo.
Primjerice, Monte Carlo metoda koristi se za:

1. Računanje integrala
$$\mathbb{E}(h(X)) = \int h(x)f(x)dx,$$
pri čemu je h realna funkcija i f funkcija gustoće slučajne varijable X. Detaljan opis nalazi se u odjeljku (5.1).

2. Računanje vjerojatnosti
$$P(Y \in A) = \mathbb{E}(\mathbb{1}_A).$$
Procjena vjerojatnosti $P(Y \in A)$ dobije se kao specijalan slučaj prethodne primjene, ako se za h uzme indikatorska funkcija, tj. $P(Y \in A)$ može se računati koristeći procjenitelj
$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_A(Y_i).$$

5.1 Monte Carlo integracija

Monte Carlo integracija je tehnika numeričke integracije koja koristi realizacije slučajne varijable iz neke distribucije za računanje konačnih integrala. To je moguće zbog činjenice da se integral može interpretirati kao matematičko očekivanje.

Definicija 5.2. *Neka je $(\Omega, \mathcal{F}, P)$ vjerojatnosni prostor i $U \sim \mathbf{U}(0,1)$. Promatramo problem procjene integrala funkcije $f : I \to \mathbb{R}$ na jediničnom intervalu $I = [0,1]$. Ako je $E|f(U)| < \infty$, tada integral*
$$\mu = \int_I f(x)dx \tag{11}$$
možemo interpretirati kao očekivanje $E[f(U)]$, odnosno
$$\mu = \int_I f(x)dx = E[f(U)]. \tag{12}$$
Neka je $U_1, U_2, \ldots, U_n$ niz nezavisnih slučajnih varijabli s uniformnom $\mathbf{U}(0,1)$ distribucijom. Monte Carlo procjenitelj dan je s
$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(U_i). \tag{13}$$

Napomena 5.3. *Primijetimo:*

1. *Za $U \sim \mathbf{U}(a,b)$ je*
$$E[f(U)] = \frac{1}{b-a}\int_a^b f(x)dx, \tag{14}$$
te je tada za $U_1, U_2, \ldots, U_n$ n.j.d. niz s $\mathbf{U}(a,b)$ distribucijom, procjenitelj za integral (14) dan s
$$\frac{b-a}{n}\sum_{i=1}^{n} f(U_i).$$

2. *Problem integracije može se generalizirati na procjenu očekivanja na skupu $I^d = [0,1]^d$ (pogledati [6]).*

Aproksimacija (13) po jakom zakonu velikih brojeva, kad $n \to \infty$, konvergira stvarnoj vrijednosti integrala.

$$\frac{f(U_1) + f(U_2) + \cdots + f(U_n)}{n} \xrightarrow{\text{(g.s)}} E[f(U)].$$

Budući da su $U_1, U_2, \ldots, U_n$ nezavisne i jednako distribuirane, $\hat{\mu}_n$ je slučajna varijabla s očekivanjem

$$\mathbb{E}(\hat{\mu}_n) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}f(U_i) = \mathbb{E}f(U) = \mu \tag{15}$$

i varijancom

$$Var(\hat{\mu}_n) = Var\Big(\frac{1}{n}\sum_{i=1}^{n} f(U_i)\Big) = \frac{1}{n^2}\sum_{i=1}^{n} Varf(U_i) = \frac{Varf(U)}{n} = \frac{\sigma^2}{n}. \tag{16}$$

Dakle, Monte Carlo procjenitelj je asimptotski nepristran za μ i konzistentan.
Jaki zakon velikih brojeva osigurava konvergenciju Monte Carlo procjene k stvarnoj vrijednosti očekivanja za velike vrijednosti n. Procjena nastala Monte Carlo metodama ima prateće uzoračke greške. Pitanje je koliki treba biti n da bi greška aproksimacije bila dovoljno mala te hoće li ta greška uopće biti mala za dani uzorak. Bitnu ulogu ocjenjivanja greške ima centralni granični teorem.
Iz centralnog graničnog teorema slijedi približna distribucija pogreške Monte Carlo procjenitelja:

$$\hat{\mu}_n - \mu \sim AN\Big(0, \frac{\sigma^2}{n}\Big),\ n \to \infty. \tag{17}$$

Očito je da će rezultat biti lošiji povećanjem varijance, dok će on biti bolji povećavanjem uzorka, odnosno s većim n.
Korijen srednje kvadratne greške (RMSE) od $\hat{\mu}_n$ je:

$$\sqrt{\mathbb{E}((\hat{\mu}_n - \mu)^2)} = \frac{\sigma}{\sqrt{n}} \tag{18}$$

što pišemo RMSE $= O(n^{-1/2})$ za $n \to \infty$ kako bi se naglasilo da je greška reda $n^{-1/2}$. Da bi se poboljšala točnost za još jednu decimalu, RMSE bi trebala biti jednu desetinu veća, što bi impliciralo 100 puta više računanja. To je jako veliki broj za samo dvije, a kamoli tri znamenke preciznijeg računa unatoč brzini $n^{-1/2}$ koju jednostavni Monte Carlo postiže.
Sasvim je jasno da jednostavni Monte Carlo nije najbolje prilagođen za probleme koji zahtijevaju jako veliku preciznost. Međutim, to ne predstavlja nedostatak jer je uglavnom dovoljna samo gruba procjena od μ. U metodama numeričke integracije postoje modeli koji imaju puno veću grešku od jednostavnog Monte Carla[10]. Većina modela ima neke idealne pretpostavke kao što su specijalni tip distribucije, gdje čak i s takvom distribucijom može doći do upotrebe netočnih

[10]Usporedbe Monte Carlo metode s različitim metodama numeričke integracije mogu se pogledati u [4].

vrijednosti parametara kao što su buduće cijene u nekom gospodarstvu ili razina potražnje. Tu je prednost pristupa jednostavnog Monte Carla jer se može računati s vrijednostima sličnijim problematici stvarnog svijeta koje se mogu dobiti njihovim približnim procjenama. Zanimljivo je i da dimenzija prostora d ne utječe na red greške. Kolika god da je dimenzija d promatranog uzorka, RMSE će i dalje biti $\sigma/\sqrt{n}$. Može se zaključiti da je jednostavni Monte Carlo u najvećoj prednosti za visoko dimenzionalne probleme koji nemaju glatku funkciju i za koje su eksplicitni oblici nedostupni.

Primjer 5.4. *Monte Carlo metodom procijenit će se integral*

$$I = \int_0^4 e^{-x} dx.$$

Ako je X slučajna varijabla s uniformnom distribucijom na $[0, 4]$, tada je

$$\mathbb{E}(f(X)) = \int_0^4 f(x)\frac{1}{4}dx = \frac{1}{4}\int_0^4 f(x)dx$$

$$\Rightarrow \int_0^4 e^{-x}dx = 4\mathbb{E}(f(X)).$$

Dakle, ako se uzme da je $X \sim \mathbf{U}(0, 4)$, onda je ovaj integral zapravo $\mathbb{E}(f(X))$ gdje je $f(x) = 4e^{-x}$.
Procjenitelj Monte Carlo metodom je:

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i),$$

gdje su x_i realizacije slučajne varijable X. Ovakva procjena ovisi o veličini uzorka n i o realizaciji uzorka.
Zadani integral pokušat će se procijeniti nekom drugom funkcijom. Uzme li se da je $X \sim \mathcal{E}xp(1)$, tada bi funkcija g bila oblika $g(x) = \mathbb{1}_{[0,4]}$, odnosno

$$E(g(X)) = \int_0^\infty \mathbb{1}_{[0,4]}e^{-x}dx = \int_0^4 e^{-x}dx.$$

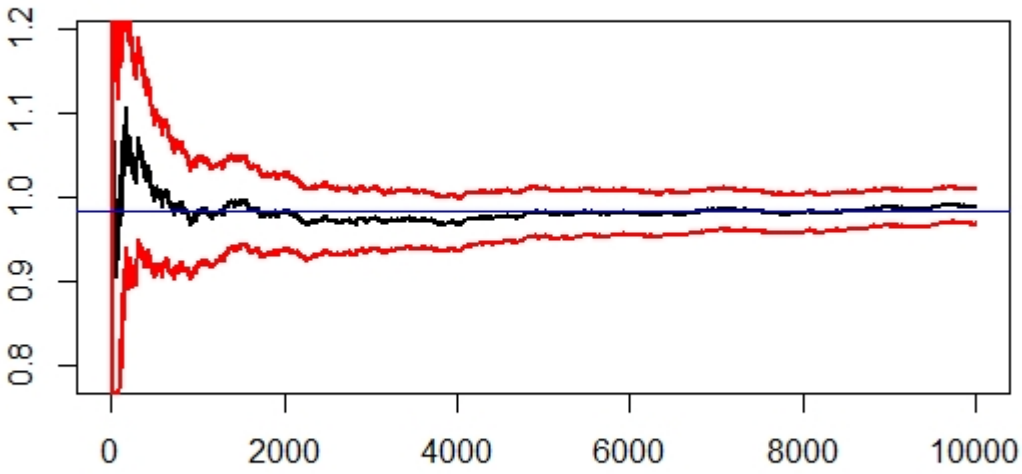
Egzaktno rješenje zadanog integrala je $e^0 - e^{-4} = 0.9816844$. Usporedba dobivenih procjena povećanjem uzorka n dana je sljedećom tablicom:

n	$\hat{I}_{\mathcal{U}(0,4)}$	$\hat{I}_{\mathcal{E}xp(1)}$
10	0.3351097	1
100	0.8956275	0.98
1000	0.9370584	0.982
10000	0.9891093	0.9818

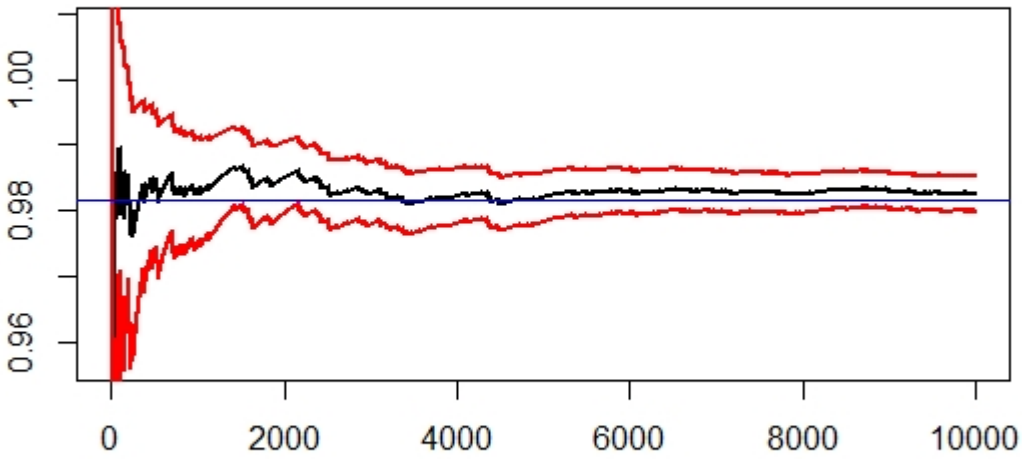
Tablica 1: Procjenitelji realizacija iz $\mathcal{U}(0, 4)$ i $\mathcal{E}xp(1)$ u ovisnosti o broju uzorka n

Realizacije generirane iz eksponencijalne distribucije daju bolju procjenu zadanog integrala već pri malom broju generiranih vrijednosti. Također, greška te procjene za $n = 10000$ *iznosi* $RMSE_{\mathcal{E}xp(1)} = 0.001300225$ *što je manje u odnosu na grešku realizacija generiranih iz uniformne distribucije za isti* n, $RMSE_{\mathcal{U}(0,4)} = 0.01012932$. *Distribucija iz koje se generiraju realizacije, bira se na način da ona što više nalikuje podintegralnoj funkciji čime će se smanjiti greška procjene.*

Pouzdani interval gdje je $n = 10000$ *iznosi* $[0.9664014,\ 1.0064898]_{\mathcal{U}(0,4)}$ *za* $\hat{\sigma}_{\mathcal{U}(0,4)} = 1.022683$, *a* $[0.9773626,\ 0.9828374]_{\mathcal{E}xp(1)}$ *za* $\hat{\sigma}_{\mathcal{E}xp(1)} = 0.1396637$ *za nivo pouzdanosti* $\alpha = 0.95$.



Slika 8: Očekivanje ± dvije standardne greške realizacija iz $\mathcal{U}(0, 4)$ za 10^4 simulacija



Slika 9: Očekivanje ± dvije standardne greške realizacija iz $\mathcal{E}xp(1)$ za 10^4 simulacija

Gornja slika predstavlja realizacije iz $\mathcal{U}(0,4)$*, a donja realizacije iz* $\mathcal{E}xp(1)$*. Na objema slikama je prikazan procjenitelj integrala* $\hat{I}_n$ *(crna linija) i raspon sandardne greške (crvene linije) za* $n = 1, 2, \ldots, 10^4$ *te stvarna vrijednost integrala* I *(plava linija).*

Izbor slučajne varijable X to jest njene distribucije i funkcije f takve da je $\mu = \mathbb{E}[f(X)]$ nije jedinstven. Cilj ih je izabrati tako da σ^2 bude što manja. Mnogo je načina[11] redukcije varijance, način koji će biti opisan u ovome radu je uzorkovanje po važnosti (*eng. Importance sampling*).

5.2 Uzorkovanje po važnosti

Uzorkovanje po važnosti jedna je od najvažnijih tehnika redukcije varijance. Sama metoda dobila je ime po funkcijama na kojima se temelji, takozvanim funkcijama važnosti. Metoda uzorkovanja po važnosti je alternativni prikaz Monte Carlo integracije. Ova tehnika posebno je korisna za procjenu vjerojatnosti rijetkih događaja[12], a koristi se da bi se smanjila varijanca Monte Carlo procjenitelja za integral

$$l = \int h(x)f(x)dx = \mathbb{E}_f(h(X)), \tag{19}$$

[11] pogledati [2], 347. *str.*

[12] Rijetkim događajima (broj događaja od interesa je puno manji od ukupnog broja događaja) smatraju se radioaktivni raspad, samoubojstva, višestruki porodi, itd.

gdje je h realna funkcija i f funkcija gustoće slučajne varijable X.

Neka je g strogo pozitivna funkcija gustoće na $\mathbb{R}$ i $h \times f \neq 0$. Tada (19) možemo pisati

$$\mathbb{E}_f(h(X)) = \int_\chi h(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_g\Big(h(X)\frac{f(X)}{g(X)}\Big),$$

gdje je χ nosač od $h \times f$.
Procjenitelj za $\mathbb{E}_g\Big(h(X)\frac{f(X)}{g(X)}\Big)$ dan je s

$$\hat{l} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)\frac{f(X_i)}{g(X_i)}, \tag{20}$$

gdje su $X_1, \ldots, X_n$ n.j.d. s gustoćom g.

Procjenitelj $\hat{l}$ naziva se **procjenitelj uzorkovanjem po važnosti** i konvergira prema $\mathbb{E}_f(h(X))$ iz istog razloga kao i regularni Monte Carlo procjenitelj $\hat{\mu}_n$, po zakonu velikih brojeva, dokle god je $supp(h \times f) \subset supp(g)$

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i)\frac{f(X_i)}{g(X_i)} \xrightarrow{\text{(g.s)}} \mathbb{E}_f(h(X)),\ n \to \infty.$$

Gustoća g naziva se **funkcija važnosti**, a $\frac{f(x_i)}{g(x_i)}$ **omjer uzorkovanja** ili **težina uzorkovanja**. Označi li se s

$$Y_i = h(X_i)\frac{f(X_i)}{g(X_i)},$$

varijanca procjenitelja (20) dana je s

$$Var\Big(\frac{1}{n}\sum_{i=1}^{n} Y_i\Big) = \frac{1}{n^2}\sum_{i=1}^{n} Var(Y_i) = \frac{Var(Y_1)}{n}.$$

Ideja je da ta varijanca bude što manja, a to se postiže odabirom funkcije važnosti g tako da Y_1 bude bliže konstanti, odnosno odabirom funkcije g koja je "blizu" f.

Teorem 5.5. ([*10*], 409. *str.*)
Izbor funkcije važnosti g koja minimizira varijancu procjenitelja $\hat{l}$ je

$$g^* = \frac{|h(x)|f(x)}{\int |h(x)|f(x)dx}.$$

Dokaz. Neka je $w = \frac{hf}{g}$, tada vrijedi

$$\begin{aligned}
\mathbb{E}_g(w^2) - \mathbb{E}_g(w)^2 &= \int w^2(x)g(x)dx - \Big(\int w(x)g(x)dx\Big)^2 \\
&= \int \frac{h^2(x)f^2(x)}{g^2(x)}g(x)dx - \Big(\int \frac{h(x)f(x)}{g(x)}g(x)dx\Big)^2 \\
&= \int \frac{h^2(x)f^2(x)}{g^2(x)}g(x)dx - \Big(\int h(x)f(x)dx\Big)^2.
\end{aligned}$$

Drugi integral ne ovisi o funkciji g, što znači da se samo prvi integral treba minimizirati. Po *Jensenovoj nejednakosti*[13] budući da je kvadratna funkcija konveksna vrijedi

$$\mathbb{E}_g(W^2) \geqslant (\mathbb{E}_g(|W|))^2 = \left(\int |h(x)|f(x)dx \right)^2,$$

što predstavlja donju granicu za $\mathbb{E}_g(W^2)$. Tvrdnja je dokazana jer je $\mathbb{E}_g*(W^2)$ jednaka toj donjoj granici, odnosno

$$\begin{aligned}
\mathbb{E}_{g^*}(W^2) &= \int \frac{h^2(x)f^2(x)}{g^{*2}(x)} g^*(x)dx = \int \frac{h^2(x)f^2(x)}{g^*(x)} dx \\
&= \int \frac{h^2(x)f^2(x)}{|h(x)|f(x)} dx \int |h(x)|f(x)dx = \int |h(x)|f(x)dx \int |h(x)|f(x)dx \\
&= \left(\int |h(x)|f(x)dx \right)^2.
\end{aligned}$$

□

Prethodni teorem ima teorijskog smisla, no u praksi, procjena optimalne funkcije važnosti g^* obično nije moguća. Dovoljno je pronaći funkciju g s težim repovima od funkcije f koja je slična $|h|f$.

Algoritam 5.6. *(Procjena uzorkovanjem po važnosti)* ([2], 362. *str.)*

1. *Odaberi funkciju važnosti g takvu da je $supp(h \times f) \subset supp(g)$*
2. *Generiraj n.j.d. niz $x_1, \ldots, x_n$ realizacija s gustoćom g i postavi $y_i = h(x_i)\frac{f(x_i)}{g(x_i)}$, za $i = 1, \ldots, n$*
3. *Procijeni l s $\hat{l} = \bar{y}$ i odredi aproksimaciju $1 - \alpha$ pouzdanog intervala*

$$\left[\bar{y} - z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}},\ \bar{y} + z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}} \right].$$

gdje je S uzoračka standardna devijacija od $y_1, \ldots, y_n$, a $z_{1-\frac{\alpha}{2}}$, $(1-\frac{\alpha}{2})$ - kvantil standardne normalne distribucije $N(0,1)$.

Primjer 5.7. *Procjena vjerojatnosti događaja $P(Z > 4.5)$, gdje je $Z \sim \mathcal{N}(0,1)$.*

Ovim primjerom će se usporediti procjena očekivanja metodom Monte Carlo integracije i metodom uzorkovanja po važnosti.
Budući da je vjerojatnost događaja jednaka očekivanju indikatora tog događaja tj.

$$P(Z > 4.5) = \mathbb{E}(\mathbb{1}_{[4.5,\infty)}),$$

[13] Jensenova nejednakost: Ako je g konveksna, tada je $\mathbb{E}g(X) \geqslant g\mathbb{E}(X)$. Ako je g konkavna, tada je $\mathbb{E}g(X) \leqslant g\mathbb{E}(X)$.

moguće je odmah primijeniti metodu integracije na očekivanje $\mu = \mathbb{E}(f(Z))$*, pri čemu je* $f(z) = \mathbb{1}_{[4.5,\infty)}$ *i* $Z \sim \mathcal{N}(0,1)$*. Procjenitelj metodom integracije za* μ *u* 1000 *simulacija je tada jednak*

$$\hat{\mu}_{10^3} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[4.5,\infty)} = 0.$$

Povećanjem broja simulacija na 10^6 *će procjenitelj biti jednak*

$$\hat{\mu}_{10^6} = 7 \times 10^{-6}.$$

Naime, pripadna je vjerojatnost uistinu jako mala jer simulirajući velik broj podataka iz standardne normalne distribucije, taj će se događaj realizirati jako malo puta što predstavlja rijedak događaj.
Pokušat će se sada procijeniti vjerojatnost $P(Z > 4.5)$ *metodom uzorkovanja po važnosti.*
Na primjer, ako se uzme u obzir distribucija s nosačem na $(4.5, \infty)$*, funkcija važnosti* g *može biti pomaknuta eksponencijalna funkcija gustoće*

$$g(y) = e^{-(y-4.5)}.$$

Tada je procjenitelj metodom uzorkovanja po važnosti za $l = \mathbb{E}(\mathbb{1}_{[4.5,\infty)})$ *u* 1000 *simulacija dan s*

$$\hat{l} = \frac{1}{n}\sum_{i=1}^{n} h(Y_i)\frac{f(Y_i)}{g(Y_i)} = \frac{1}{n}\sum_{i=1}^{n} \frac{e^{-\frac{y_i^2}{2}+y_i-4.5}}{\sqrt{2\pi}},$$

pri čemu su $y_1, \ldots, y_n$ *realizacije n.j.d. niza iz* g *i* $h(y) = 0$ *za* $y \leqslant 4.5$ *i* $h(y) = 1$ *za* $y > 4.5$*.*

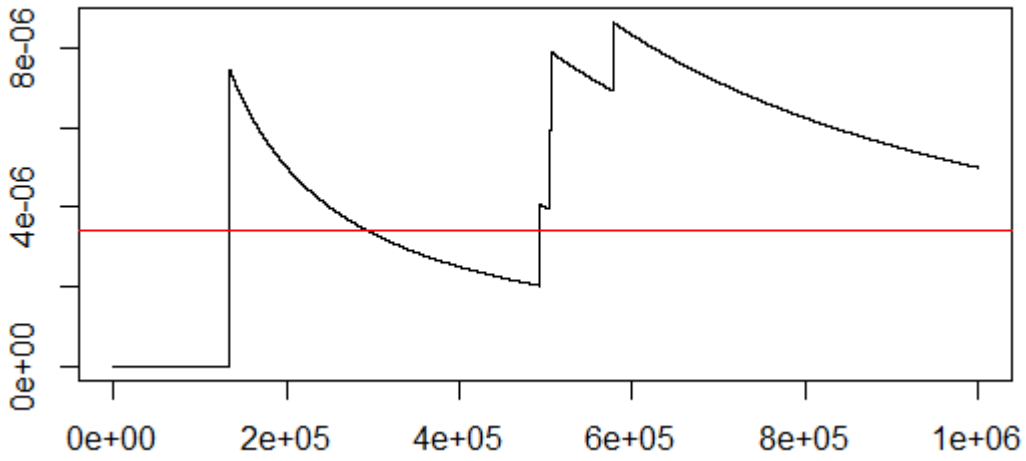
Stvarna vrijednost vjerojatnosti događaja $P(Z > 4.5)$ *iznosi* 3.397673×10^{-6}*. Tablicom su prikazani dobiveni procjenitelji navedenim metodama te njihove greške.*

	Monte Carlo integracija	**Uzorkovanje po važnosti**
	$\hat{\mu}$	$\hat{l}$
$n = 10^3$	0	3.413944×10^{-6}
$n = 10^6$	7×10^{-6}	3.398081×10^{-6}
$RMSE$ *za* $n = 10^3$	-	1.390751×10^{-8}
$RMSE$ *za* $n = 10^6$	2.645743×10^{-5}	4.414204×10^{-9}

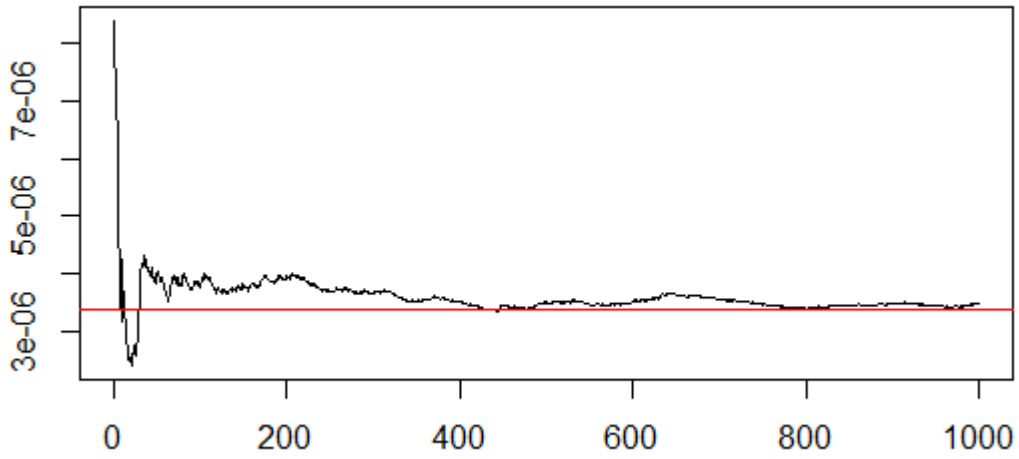
Tablica 2: Usporedba procjenitelja dobivenih metodom integracije i uzorkovanjem po važnosti u ovisnosti o broju uzorka n

Greška metode uzorkovanja po važnosti je puno manja u odnosu na metodu integracije, samim time je i procjena točnija. Generalno, korištenje metode Monte Carlo integracije za procjenu očekivanja repnih događaja standardne normalne distribucije nije preporučljivo. Sam je proces

spor zbog jako velikog broja iteracija i najčešće se kao rezultat dobije vrijednost 0. *Na primjer, računanje vjerojatnosti* $P(Z > 20)$ *za* $Z \sim \mathcal{N}(0,1)$ *simuliranjem realizacija iz* $\mathcal{N}(0,1)$ *nije moguće metodom Monte Carlo integracije.*



Slika 10: Procjenjena vjerojatnost događaja $P(Z > 0)$, $Z \sim \mathcal{N}(0,1)$ simuliranjem realizacija iz $\mathcal{N}(0,1)$ za $n = 10^6$ metodom Monte Carlo integracije



Slika 11: Procjenjena vjerojatnost događaja $P(Z > 0)$, $Z \sim \mathcal{N}(0,1)$ simuliranjem realizacija iz pomaknute eksponencijalne distribucije za $n = 10^3$ metodom uzorkovanja po važnosti

Crvena linija na obije slike predstavlja stvarnu vjerojatnost $P(Z > 4.5)$*, dok je crna linija njena aproksimacija dobivena metodom Monte Carlo integracije, odnosno uzorkovanjem po važnosti. Iz gornjeg prikaza se vidi koliko je rijedak promatrani događaj. Aproksimacija se počinje približavati stvarnoj vrijednosti tek u više od* 10^6 *iteracija, dok je u slučaju s metodom uzorkovanja ta aproksimacija približna stvarnoj vrijednosti već u* $n = 500$ *iteracija.*

Prednost metode uzorkovanja po važnosti je manja greška u odnosu na grešku nastalu prilikom procjene očekivanja metodom Monte Carlo integracije. Također, pogodnija je za događaje koji se realiziraju s jako malom vjerojatnošću te za simuliranje vrijednosti iz kompleksnih distribucija. Odabirom prikladne funkcije važnosti povećavaju se vjerojatnosti pojavljivanja tih događaja jer se generira veći broj „značajnijih“ realizacija iz slučajne varijable s distribucijom g. Za procjenu joj je dovoljno koristiti manji broj uzoraka što rezultira većom brzinom simulacije.

Na kraju rada će se opisati primjer višedimenzionalnog integrala čije se rješenje ne može dobiti analitičkim putem. Primjer će se riješiti Monte Carlo integracijom opisanom u ovome radu te dvama ugrađenim funkcijama u R-u, od kojih jedna također koristi Monte Carlo metodu temeljenu na uzorkovanju po važnosti, a druga je numerička metoda temeljena na adaptivnoj višedimenzionalnoj integraciji.

Primjer 5.8. *Aproksimacija integrala*

$$I = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{|x_1 + x_2 + x_3|}e^{-(x_1^2+x_2^2+x_3^2)/2}dx_1dx_2dx_3.$$

Ako se definira slučajna varijabla Y takva da je $Y = \sqrt{|X_1 + X_2 + X_3|}(2\pi)^{3/2}$, gdje su $X_1, X_2, X_3 \overset{n.j.d.}{\sim} \mathcal{N}(0,1)$, tada se procjena integrala dobiva očekivanjem $\mathbb{E}(Y)$.

Također, navedeni integral izračunat će se i s funkcijama ugrađenim u R-u. Funkcije za računanje integrala visokih dimenzija zovu se suave() i adaptIntegrate(). Dobiveni rezultati prikazani su tablicom.

	Monte Carlo integracija	**suave()**	**adaptIntegrate()**
procjena integrala	16.90623	17.04174	17.04185
greška	0.01363362	0.001975457	0.0001704183

Tablica 3: Usporedba Monte Carlo integracije s funkcijama u R-u

Funkcija suave() iz R-a je zapravo implementirana Monte Carlo metoda temeljena na metodi uzorkovanja po važnosti. Sve tri metode su procijenile integral na osnovu $n = 10^3$ simulacija. Tablični rezultati promatranjem greške, pokazuju najbolju procjenu implementiranom funkcijom adaptIntegrate() temeljenu na numeričkoj analizi.
Promatrani problem je trodimenzionalan, postavljanjem istog problema na pet dimenzija[14]*, funkcijom adaptIntegrate() se ne može riješiti problem, dok ga Monte Carlo metode rješavaju s lakoćom ($\hat{I} = 121.6571$).*

[14] $I = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{|x_1 + x_2 + x_3 + x_4 + x_5|}e^{-(x_1^2+x_2^2+x_3^2+x_4^2+x_5^2)/2}dx_1dx_2dx_3dx_4dx_5.$

Literatura

[1] Benšić M., Šuvak N., Uvod u vjerojatnost i statistiku, Sveučilište J. J. Strossmayera, Odjel za matematiku, Osijek, 2014.

[2] Botev Z. I., Kroese D. P., Taimre T., Handbook of Monte Carlo Methods, John Wiley & Sons, Inc., New Jersey, 2011.

[3] Casella G., Robert C. P., Introducing Monte Carlo Methods with R, Springer-Verlag, New York, 2010.

[4] Jones O., Maillardet R., Robinson A., Introduction to Scientific Programming and Simulation Using R, Chapman & Hall/CRC, Boca Raton, 2009.

[5] Kroese D. P., Rubinstein R. Y., Simulation and the Monte Carlo Method, Third Edition, John Wiley & Sons, Inc., New Jersey, 2017.

[6] Leobacher G., Pillichshammer F., Introduction to Quasi-Monte Carlo Integration and Applications, Compact Textbooks in Mathematics, Birkhäuser, Basel, 2014.

[7] Owen A. B., Monte Carlo theory, methods and examples, Stanford University, Department of Statistics, 2013.

[8] Pauše Ž., Uvod u matematičku statistiku, Školska knjiga, Zagreb, 1993.

[9] Sarapa N., Teorija vjerojatnosti, Školska knjiga, Zagreb, 1987.

[10] Wasserman L., All of Statistics - A Concise Course in Statistical Inference, Springer Texts in Statistics, New York, 2004.

Sažetak

Monte Carlo metode su bilo koji matematički modeli i algoritmi čija je glavna značajka korištenje velikog broja slučajnih brojeva koji se simuliraju nasumičnim uzorkovanjem. Izuzetno su prikladne za rješavanje širokog spektra složenih problema u različitim poljima znanosti.
U prvom dijelu rada opisani su načini generiranja realizacija nizova nezavisnih slučajnih varijabli, čija se kvaliteta ispituje statističkim testovima. Drugim dijelom dan je uvid u dvije različite metode generiranja realizacija slučajnih varijabli, metode inverzije i metode odbacivanja. Nakon toga, s ciljem procjene matematičkih izraza poput konačnih integrala, objašnjena je metoda Monte Carlo integracije. Na kraju je opisana tehnika redukcije varijance uzorkovanja po važnosti kojom se procjenjuju rijetki događaji.

Ključne riječi: uzorkovanje, simulacije, Monte Carlo metode, procjena, integracija

Summary

Monte Carlo methods are any mathematical models and algorithms whose main feature is the usage of large number of random numbers that are simulated by random sampling. They are extremely suitable for solving a wide range of complex problems in various fields of science.
The first part of the paper describes the ways of generating the realization of a series of independent random variables, the quality of which is examined by statistical tests. The second part provides an insight into two different methods of generating the realization of random variables, the inversion method and the rejection method. Subsequently, in order to estimate mathematical expressions such as finite integrals, the Monte Carlo integration method is explained. Finally, variance reduction technique called importance sampling used for estimating rare events is described.

Key words: sampling, simulation, Monte Carlo methods, estimation, integration

Životopis

Rođena sam 13.veljače, 1995. godine u Osijeku. Nakon završene Osnovne škole Franje Krežme u Osijeku upisujem Isusovačku klasičnu gimnaziju s pravom javnosti u Osijeku. Gimnaziju završavam 2014. godine te iste godine upisujem preddiplomski studij matematike na Odjelu za matematiku u Osijeku. Preddiplomski studij završavam 2017. godine sa završnim radom na temu Zakon velikih brojeva te time stječem akademski stupanj prvostupnice matematike. Iste godine upisujem diplomski studij na Odjelu za matematiku u Osijeku, smjer Financijska matematika i statistika. U rujnu 2019. godine odrađujem stručnu praksu u tvrtki Osijek - Koteks d.d.