

# Monte Carlo i Markovljevi lanci

---

Kopić, Katarina

Master's thesis / Diplomski rad

2023

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://um.nsk.hr/um:nbn:hr:126:892899>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-31**



**mathos**

Repository / Repozitorij:

[Repository of School of Applied Mathematics and Informatics](#)





SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ODJEL ZA MATEMATIKU

Sveučilišni diplomski studij matematike  
smjer: Matematika i računarstvo

# **Monte Carlo Markovljevi lanci**

DIPLOMSKI RAD

**Katarina Kopic**

Osijek, 2023.



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ODJEL ZA MATEMATIKU

Sveučilišni diplomski studij matematike  
smjer: Matematika i računarstvo

# Monte Carlo Markovljevi lanci

DIPLOMSKI RAD

**Katarina Kopic**

Mentor:

**izv. prof. dr. sc. Nenad Šuvak**

Osijek, 2023.



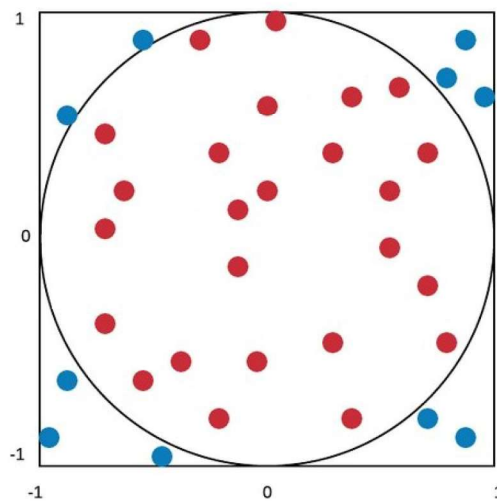
# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Monte Carlo metoda</b>	<b>3</b>
2.1	Osnove Monte Carlo metode i primjene . . . . .	3
<b>3</b>	<b>Monte Carlo Markovljevi lanci</b>	<b>5</b>
3.1	Markovljevi lanci . . . . .	5
3.2	Metropolis-Hastings algoritam . . . . .	15
3.2.1	Opis algoritma . . . . .	15
3.2.2	Specijalni slučajevi Metropolis-Hastings algoritma . . . . .	16
3.3	Gibbsovo uzorkovanje . . . . .	20
<b>4</b>	<b>Primjena MCMC metode u obradi slika</b>	<b>23</b>
4.1	Isingov model . . . . .	23
4.2	Redukcija šuma na slici pomoću MCMC metode . . . . .	24
<b>5</b>	<b>Kod</b>	<b>29</b>
	<b>Literatura</b>	<b>33</b>
	<b>Sažetak</b>	<b>35</b>
	<b>Summary</b>	<b>37</b>
	<b>Životopis</b>	<b>39</b>



# 1 | Uvod

Monte Carlo metoda nastala je tijekom Drugog svjetskog rata kako bi se poboljšalo donošenje odluka u neizvjesnim uvjetima, a popularizirali su je *Stanislaw Ulam*, *Enrico Fermi*, *John von Neumann* i *Nicholas Metropolis*. Metoda predstavlja način rješavanja problema koji se oslanja na generiranje velikog broja slučajnih brojeva, a uobičajeni primjer korištenja metode jest aproksimacija vrijednosti broja  $\pi$ . Pretpostavimo da imamo krug polumjera jedan upisan u kvadrat kao na slici 1.1. Omjer površine kruga i kvadrata sa slike iznosi  $\frac{\pi}{2^2} = \frac{\pi}{4}$ . Uniformno odabiremo



Slika 1.1: Aproksimacija broja  $\pi$  pomoću Monte Carlo metode

točke iz pravokutnika  $[-1, 1] \times [-1, 1]$  i izračunamo omjer točaka koje se nalaze unutar kruga i ukupnog broja točaka. Za dovoljno veliki broj generiranih točaka omjer bi trebao približno iznositi  $\frac{\pi}{4}$ . Ako pomnožimo to s 4 dobit ćemo aproksimaciju broja  $\pi$ .

Također, metoda se koristi za bolje razumijevanje slučajnosti u raznim područjima kao što su umjetna inteligencija, cijene dionica, predviđanje prodaje, upravljanje projektima, itd. Kao motivaciju za primjenu Monte Carlo metode promotrimo jednostavan primjer.

## Primjer 1.

Neka je zadana funkcija  $f : \mathbb{R} \rightarrow [-1, 1]$ ,  $f(x) = \sin(x)$ . Izračunajmo površinu ispod grafa funkcije na intervalu  $[0, \pi]$  egzaktno i korištenjem Monte Carlo metode, gdje je broj ponavljanja  $N \in \{100, 1000, 10000\}$ .

**Rješenje:**

Odredimo najprije egzaktnu vrijednost površine ispod grafa funkcije:

$$\begin{aligned} & \int_0^{\pi} \sin(x) dx \\ &= -\cos(x) \Big|_{x=0}^{\pi} \\ &= 2. \end{aligned}$$

Gornji integral možemo aproksimirati koristeći Monte Carlo metodu. Uniformno odabiremo  $N$  točaka iz pravokutnika  $[0, \pi] \times [0, 1]$ . Neke od točaka će biti ispod, a neke iznad grafa funkcije  $f$ . Aproksimaciju površine tada možemo izraziti na sljedeći način:

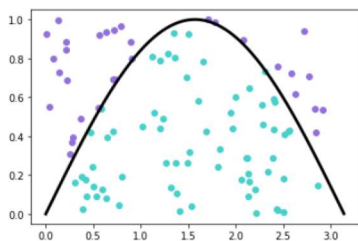
$$\frac{\text{Broj točaka ispod grafa}}{\text{Ukupan broj točaka}} \cdot \text{Površina pravokutnika } [0, \pi] \times [0, 1].$$

Površina pravokutnika u našem slučaju iznosi  $\pi$ , a broj generiranih točaka te aproksimacije površine ispod grafa funkcije možemo vidjeti u Tablici 1.1.

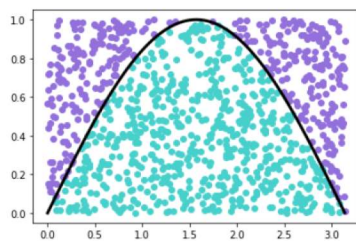
Br. ponavljanja	100	1000	10000
Br. točaka iznad grafa funkcije	32	359	3607
Br. točaka ispod grafa funkcije	68	641	6393
Površina ispod grafa funkcije	2.136	2.014	2.008

Tablica 1.1: Rezultati Monte Carlo simulacija za  $N \in \{100, 1000, 10000\}$

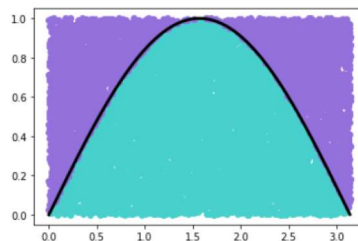
Na sljedećim grafičkim prikazima možemo vidjeti funkciju  $f(x) = \sin(x)$  na intervalu  $[0, \pi]$  te nasumično generirane točke za  $N=100$ ,  $N=1000$  i  $N=10000$ . Intuitivno možemo naslutiti da je s povećanjem broja generiranih točaka aproksimacija površine bliža stvarnoj vrijednosti površine ispod grafa funkcije  $f$ .



Slika 1.2:  $N=100$



Slika 1.3:  $N=1000$



Slika 1.4:  $N=10000$

Jedan posebno popularan podskup Monte Carlo metoda poznat je kao Monte Carlo Markovljevi lanci (MCMC), koji pruža jednostavan i intuitivan način za simulaciju vrijednosti iz nepoznate distribucije te korištenje simuliranih vrijednosti za izvođenje naknadnih analiza. To omogućava da bude primjenjiv u velikom broju domena o čemu će više riječi biti u nastavku rada.



## 2 | Monte Carlo metoda

### 2.1 Osnove Monte Carlo metode i primjene

Monte Carlo metoda (poznata i kao Monte Carlo simulacije) koristi se uglavnom u tri klase problema: optimizacija, numerička integracija i generiranje uzorka iz vjerojatnosne distribucije. Riječ je o metodi koja pomoću slučajno generiranih vrijednosti i velikog broja ponavljanja opisuje ponašanje složenih sustava.

Svoje ime duguje poznatom kasinu smještenom u Kneževini Monako, a uveli su ga *Stanislaw Ulam*, *John von Neumann* i *Nicholas Metropolis* 1946. godine. Iste godine, pojavom prvog elektroničkog računala ENIAC, dolazi do detaljnog proučavanja metode.

Ima široku lepezu primjena u područjima u kojima je jasna i očita prisutnost teško mjerljivih slučajnih utjecaja, posebice u poslovanju i ulaganju. Koristi se za procjenu vjerojatnosti prekoračenja troškova u velikim projektima i vjerojatnosti da će se cijena imovine promijeniti na određeni način. Telekomu ju koriste za procjenu performansi mreže u različitim scenarijima, što im pomaže da optimiziraju svoje mreže. Monte Carlo simulacije imaju mnoge primjene izvan poslovanja i financija, kao što su meteorologija, astronomija i fizika čestica. Više o primjenama Monte Carlo metode možete pronaći u [17].

Prema [12] Monte Carlo simulacije u osnovi imaju četiri jednostavna koraka:

- **Određivanje jednadžbe prijenosa**  
Za izradu Monte Carlo simulacije potreban nam je model problema kojeg želimo analizirati. Matematički izraz za model naziva se "jednadžba prijenosa";
- **Definiranje ulaznih parametara**  
Za svaki faktor u jednadžbi prijenosa trebamo odrediti iz koje distribucije dolazi;
- **Simulacija**  
Potrebno je generirati veliki skup slučajno odabranih vrijednosti;
- **Analiza izlaznih podataka**  
Simulirane podatke uvrstiti u jednadžbu prijenosa.



## 3 | Monte Carlo Markovljevi lanci

Nakon što smo ukratko predstavili ideju Monte Carlo metode, u ovom poglavlju ćemo razmotriti specijalni slučaj ove metode, tj. Monte Carlo Markovljevi lanci metodu (ili, kraće, MCMC). Problem kod primjene klasične Monte Carlo metode javlja se kod generiranja uzoraka iz složenih vjerojatnosnih distribucija. Pokušaji rješavanja ovog problema korijeni su MCMC metode, tj. metode uzorkovanja iz dane distribucije. To je familija algoritama koji koriste Markovljeve lance za izvođenje Monte Carlo metode. Postoje različiti algoritmi za konstruiranje lanaca, uključujući Metropolis–Hasting algoritam te Gibbsovo uzorkovanje, a njihov princip rada bit će predstavljen u nastavku. Kako bismo lakše razumjeli navedenu metodu, najprije ćemo definirati neke važne pojmove i iskazati ključne teoreme vezane uz teoriju Markovljevih lanaca.

### 3.1 Markovljevi lanci

#### Definicija 1.

Neka je  $(\Omega, \mathcal{F}, \mathcal{P})$  vjerojatnosni prostor. Funkciju  $\mathbb{X} : \Omega \rightarrow \mathbb{R}^n$  za koju je skup  $\{\omega \in \Omega : \mathbb{X}(\omega) \in B\}$  događaj u  $\mathcal{F}$ , tj.  $\mathbb{X}^{-1}(B) \in \mathcal{F}$ , za svaki  $B \in \mathcal{B}^n(\mathbb{R})$ , zovemo *n-dimenzionalan slučajni vektor* (ili, kraće, *slučajni vektor*).

#### Definicija 2.

Neka je  $S$  diskretan skup stanja. Slučajni proces  $X = (X_n, n \in \mathbb{N}_0)$  na vjerojatnosnom prostoru  $(\Omega, \mathcal{F}, \mathcal{P})$  s vrijednostima u skupu stanja  $S$  zovemo **Markovljev lanac** ako vrijedi:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i), \quad (3.1)$$

$\forall n \in \mathbb{N}_0$  i za svaki izbor stanja  $i_0, i_1, \dots, i_{n-1}, i, j \in S$ , za koje su gornje vjerojatnosti dobro definirane.

Svojstvo (3.1) naziva se **Markovljevo svojstvo** (ili, kraće, MS) i kaže da su, uvjetno na poznato sadašnje stanje, prošlost i budućnost procesa nezavisne.

Markovljev lanac zadajemo funkcijom prijelaznih vjerojatnosti što je vidljivo u sljedećoj definiciji.

#### Definicija 3.

Za stanja  $i, j \in S$  i  $0 \leq s < t$ ,  $s, t \in \mathbb{N}_0$ , funkcija prijelaznih vjerojatnosti definirana je pravilom

$$p(i, s; t, j) = P(X_t = j \mid X_s = i).$$

Funkcija 1–koračnih prijelaznih vjerojatnosti je tada definirana s

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

Odgovor na pitanje što je dovoljno za potpuno opisivanje dinamike Markovljevog lanca daje sljedeći teorem.

**Teorem 1.**

*Markovljev lanac u potpunosti je određen poznavanjem distribucije od  $X_0$  te funkcije 1–koračnih prijelaznih vjerojatnosti.*

*Dokaz.*

Označimo s  $\lambda_i = P(X_0 = i)$ ,  $i \in S$ . Tada je  $\lambda = (\lambda_i, i \in S)$  distribucija slučajne varijable  $X_0$ , tj. **početna distribucija Markovljevog lanca**. Također, poznate su nam i 1–koračne prijelazne vjerojatnosti, tj.  $p(i, n; n + 1, j) = P(X_{n+1} = j \mid X_n = i)$ .

Pokažimo da je poznavanje početne distribucije Markovljevog lanca i 1–koračnih prijelaznih vjerojatnosti dovoljno kako bismo odredili proizvoljnu konačnodimenzionalnu distribuciju Markovljevog lanca.

Za proizvoljna stanja  $i_0, \dots, i_n \in S$  i proizvoljni  $n \in \mathbb{N}$  vrijedi:

$$\begin{aligned} &P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i_n) = \\ &= P(X_n = i_n \mid X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) \cdot P(X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) \\ &\stackrel{\text{MS}}{=} P(x_n = i_n \mid X_{n-1} = i_{n-1}) \cdot P(X_{n-1} = i_{n-1} \mid X_{n-2} = i_{n-2}, \dots, X_0 = i_0) \cdot \\ &\quad \cdot P(X_{n-2} = i_{n-2}, \dots, X_0 = i_0) \\ &\stackrel{\text{MS}}{=} p(i_{n-1}, n - 1; n, i_n) \cdot p(i_{n-2}, n - 2; n - 1, i_{n-1}) \cdot \dots \cdot p(i_0, 0; 1, i_1) \cdot P(X_0 = i_0) \\ &= p(i_{n-1}, n - 1; n, i_n) \cdot p(i_{n-2}, n - 2; n - 1, i_{n-1}) \cdot \dots \cdot p(i_0, 0; 1, i_1) \cdot \lambda_{i_0}. \end{aligned}$$

□

Ako 1–koračne prijelazne vjerojatnosti ne ovise o vremenu, tj. ako vrijedi

$$p(i, n; n + 1, j) = p(i, m; m + 1, j), \forall n, m \in \mathbb{N}_0, n \neq m,$$

takve Markovljeve lance nazivamo (**vremenski**) **homogeni Markovljevi lanci**.

Vjerojatnosti  $P(X_{n+1} = j \mid X_n = i) = p(i, n; n + 1, j) = p_{ij}$  možemo organizirati u matricu:

$$\begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} & \dots \\ p_{21} & p_{22} & \dots & p_{2n} & \dots \\ \vdots & \vdots & & \vdots & \\ p_{n1} & p_{n2} & \dots & p_{nn} & \dots \\ \vdots & \vdots & & \vdots & \end{bmatrix},$$

a nazivamo ju **matrica 1–koračnih prijelaznih vjerojatnosti**.

**Definicija 4.**

Matrica  $\Pi = (p_{ij})_{i,j \in S}$  je **stohastička** i ima sljedeća svojstva:

$$(i) \quad p_{ij} \geq 0, \quad \forall i, j \in S;$$

$$(ii) \quad \sum_{j \in S} p_{ij} = 1.$$

Markovljev lanac s početnom distribucijom  $\lambda$  i matricom 1–koračnih prijelaznih vjerojatnosti  $\Pi$  označavamo kao  $(\lambda, \Pi)$  Markovljev lanac. Ireducibilni  $(\lambda, \Pi)$  Markovljevi lanci od posebne su važnosti za ergodski teorem (kojega ćemo kasnije iskazati). Ergodski teorem je granični rezultat koji kaže da je za gotovo sve trajektorije ireducibilnog pozitivno povratnog Markovljevog lanca granično vremensko usrednjenje jednako prostornom usrednjenju. Kako bismo definirali ireducibilnost lanca, najprije ćemo definirati dostižnost i komuniciranje stanja.

**Definicija 5.**

Kažemo da je stanje  $j \in S$  dostižno iz stanja  $i \in S$  (pišemo,  $i \rightarrow j$ ) ako je

$$P(T_j < \infty \mid X_0 = i) > 0,$$

gdje je  $T_j = \min\{n \in \mathbb{N}_0 : X_n = j\}$  **vrijeme prvog posjeta** Markovljevog lanca stanju  $j \in S$ .

Gornja definicija nam kaže da je stanje  $j$  dostižno iz stanja  $i$ , ako je vjerojatnost da je Markovljev lanac koji je startao iz stanja  $i$  prvi put posjetio stanje  $j$  u konačnom vremenu, veća od nule.

**Definicija 6.**

Za stanja  $i, j \in S$  kažemo da komuniciraju ako je stanje  $i$  dostižno iz stanja  $j$  ( $j \rightarrow i$ ) te stanje  $j$  dostižno iz stanja  $i$  ( $i \rightarrow j$ ) i pišemo  $i \leftrightarrow j$ .

**Definicija 7.**

Markovljev lanac je **ireducibilan** ako se skup stanja  $S$  sastoji od samo jedne klase komuniciranja, tj. ako sva stanja međusobno komuniciraju.

**Definicija 8.**

Neka je  $(X_n, n \in \mathbb{N}_0)$   $(\lambda, \Pi)$  Markovljev lanac sa skupom stanja  $S$ . **Vrijeme prvog povratka** lanca u stanje  $i \in S$  je:

$$T_i^{(1)} = \min\{n \in \mathbb{N} : X_n = i\}.$$

Stanja možemo klasificirati na povratna i prolazna o čemu nam govori sljedeća definicija.

**Definicija 9.**

Za stanje  $i \in S$  kažemo da je **povratno** ako je

$$P(T_i^{(1)} < \infty \mid X_0 = i) = 1.$$

Ako vrijedi

$$P(T_i^{(1)} < \infty \mid X_0 = i) < 1,$$

onda za stanje  $i \in S$  kažemo da je **prolazno**.

Dakle, za stanje  $i \in S$  kažemo da je povratno ako se Markovljev lanac u njega vraća beskonačno mnogo puta, a ako se ne vraća beskonačno mnogo puta kažemo da je prolazno.

U nastavku rada zanimat će nas posebna klasa slučajnih procesa, a riječ je o stacionarnim procesima u užem smislu ili strogo stacionarnim procesima.

#### Definicija 10.

Za slučajni proces  $(X_n, n \in \mathbb{N}_0)$  na vjerojatnosnom prostoru  $(\Omega, \mathcal{F}, P)$  kažemo da je **strogo stacionaran** ako su za sve  $m, n \in \mathbb{N}_0$  slučajni vektori  $(X_0, X_1, \dots, X_m)$  i  $(X_n, X_{n+1}, \dots, X_{n+m})$  jednako distribuirani.

#### Definicija 11.

Neka je  $(X_n, n \in \mathbb{N}_0)$   $(\lambda, \Pi)$  Markovljev lanac sa skupom stanja  $S$ . Za vjerojatnosnu distribuciju  $\pi = (\pi_k, k \in S)$  na skupu  $S$  kažemo da je stacionarna distribucija Markovljevog lanca, ako vrijedi

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

tj. ako po komponentama vrijedi

$$\pi_i = \sum_{k \in S} \pi_k p_{ki}, \quad \forall i \in S.$$

Primijetimo da, ako stacionarna distribucija Markovljevog lanca postoji, vrijedi

$$\pi\Pi^n = (\pi\Pi) \cdot \Pi^{n-1} = \pi\Pi^{n-1} = \dots = \pi,$$

tj. jednodimenzionalne distribucije lanca se ne mijenjaju. Također, ne mijenjaju se ni njegove konačnodimenzionalne distribucije.

#### Definicija 12.

Za stanje  $i \in S$  kažemo da je **pozitivno povratno** ako je  $E_i[T_i^{(1)}] < \infty$ .

#### Definicija 13.

Neka je  $(X_n, n \in \mathbb{N}_0)$  Markovljev lanac sa skupom stanja  $S$  i matricom 1-koračnih prijelaznih vjerojatnosti  $\Pi$ . Za vjerojatnosnu distribuciju  $\pi = (\pi_k, k \in S)$  kažemo da je **granična distribucija** Markovljevog lanca ako vrijedi

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

Definicija 13 kaže da nakon dovoljno mnogo koraka nije važno koje je početno stanje lanca, već će se lanac bez obzira na to naći u stanju  $j$  s vjerojatnošću  $\pi_j$ . Ovo svojstvo je bitno kod MCMC metoda jer odabir početnog stanja nema utjecaj na postizanje granične distribucije Markovljevog lanca.

#### Definicija 14.

Neka je  $(X_n, n \in \mathbb{N}_0)$   $(\lambda, \Pi)$  Markovljev lanac. Za stanje  $i \in S$  s  $d(i)$  označavamo najveći zajednički djelitelj skupa  $\{n \in \mathbb{N} : p_{ii}^{(n)} > 0\}$ , pri tome ako je skup prazan vrijedi  $d(i) = 1$ . Ako je  $d(i) = 1$  kažemo da je stanje  $i$  **aperiodično**, u suprotnom kažemo da je periodično s periodom  $d(i)$ .

Prema [3, str. 132, str. 395], za stanje Markovljevog lanca kažemo da je **ergodsko** ako je aperiodično i pozitivno povratno, a ireducibilan Markovljev lanac čija su sva stanja ergodska zovemo **ergodski**.

**Teorem 2.**

Neka je  $(X_n, n \in \mathbb{N}_0)$  ireducibilan Markovljev lanac sa skupom stanja  $S$  i matricom 1–koračnih prijelaznih vjerojatnosti  $\Pi$ . Sljedeće tvrdnje su ekvivalentne:

- (i) sva stanja u lancu su pozitivno povratna;
- (ii) postoji stanje  $i \in S$  koje je pozitivno povratno;
- (iii) Markovljev lanac ima stacionarnu distribuciju  $\pi$ .

Dokaz teorema može se pronaći u [5] (Poglavlje 7, Teorem 7.14).

**Teorem 3.**

Neka je  $(X_n, n \in \mathbb{N}_0)$  Markovljev lanac sa skupom stanja  $S$  i matricom 1–koračnih prijelaznih vjerojatnosti  $\Pi$  koji je ireducibilan i aperiodičan te ima stacionarnu distribuciju  $\pi$ . Tada je

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

te specijalno vrijedi

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

tj. stacionarna distribucija lanca ujedno je i granična distribucija.

Dokaz teorema može se pronaći u [5] (Poglavlje 8, Teorem 8.9).

Dakle, Teorem 2 tvrdi da egzistencija stacionarne distribucije ireducibilnog Markovljevog lanca povlači pozitivnu povratnost lanca. Dodatno, ako je Markovljev lanac aperiodičan onda nam Teorem 3 kaže da je stacionarna distribucija jednaka graničnoj.

O graničnom ponašanju prosjeka (tj. aritmetičke sredine) za niz nezavisnih i jednako distribuiranih slučajnih varijabli koje imaju očekivanje govore nam slabi i jaki zakoni velikih brojeva (vidi [4, str. 391., str. 413.]). Sljedeći teorem poznat pod nazivom *Ergodski teorem* svojevrsna je generalizacija jakog zakona velikih brojeva za Markovljeve lance. Govori o graničnom ponašanju srednjih vrijednosti vezanih uz Markovljeve lance kroz vrijeme.

**Teorem 4. (Ergodski teorem)**

Neka je  $(X_n, n \in \mathbb{N}_0)$  pozitivno povratan i ireducibilan Markovljev lanac te neka je  $\pi = (\pi_k, k \in S)$  njegova jedinstvena stacionarna distribucija. Neka je  $f : S \rightarrow \mathbb{R}$  ograničena ili nenegativna funkcija. Tada vrijedi:

$$P \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} f(X_i) = \sum_{k \in S} f(k) \pi_k \right) = 1.$$

**Napomena 1.**

- 1) Uočimo da je izraz  $\sum_{k \in S} f(k) \pi_k$  dobro definiran za konačan skup  $S$ . Za prebrojiv skup  $S$  i nenegativnu funkciju  $f$  imamo red nenegativnih realnih brojeva koji ili divergira u  $\infty$  ili konvergira. Ako je  $S$  prebrojiv i funkcija  $f$  ograničena, tj.  $|f(k)| \leq K, \forall k \in S$ , imamo

$$\sum_{k \in S} |f(k)| \pi_k \leq K \sum_{k \in S} \pi_k = K,$$

pa se radi o apsolutno konvergentnom redu.

Zbog jednostavnosti, u nastavku rada uvodimo oznaku:

$$\sum_{k \in S} f(k) \pi_k = \pi(f).$$

- 2) Veličinu  $\pi(f)$  možemo prikazati na drugačiji način. U skladu s definiranim  $\pi_j$  (vidi [5, Poglavlje 7, str. 43.]) slijedi:

$$\begin{aligned} \pi(f) &= \sum_{j \in S} f(j) \pi_j \\ &= \sum_{j \in S} f(j) \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{n=0}^{T_i^{(1)}-1} I_{\{X_n=j\}} \right] \\ &= \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{j \in S} f(j) \sum_{n=0}^{T_i^{(1)}-1} I_{\{X_n=j\}} \right] \\ &= \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{n=0}^{T_i^{(1)}-1} \left( \sum_{j \in S} f(j) I_{\{X_n=j\}} \right) \right] \\ &= \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{n=0}^{T_i^{(1)}-1} f(X_n) \right]. \end{aligned} \tag{3.2}$$

*Dokaz teorema.*

Pretpostavimo da vrijedi

$$P_i \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) = 1, \forall i \in S.$$

Ako vrijedi gornja pretpostavka, tada je:

$$\begin{aligned} &P \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) = \\ &\stackrel{\text{FPV}}{=} \sum_{i \in S} P \left( \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) \middle| X_0 = i \right) \cdot P(X_0 = i) \end{aligned}$$



$$\begin{aligned}
&= \sum_{i \in S} P_i \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) \cdot \pi_i \\
&= \sum_{i \in S} \pi_i \cdot 1 \\
&= 1.
\end{aligned}$$

Dakle, dovoljno je dokazati da vrijedi

$$P_i \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) = 1, \forall i \in S.$$

Odaberimo proizvoljno stanje  $i \in S$  i pokažimo da je za njega

$$P_i \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \sum_{j \in S} f(j) \pi_j \right) = 1.$$

Pretpostavimo da je  $f : S \rightarrow \mathbb{R}$  nenegativna (dozvoljeno je  $\sum_{j \in S} f(j) \pi_j = \infty$ ). Pri-

sjetimo se,  $T_i^{(k)}, k \geq 1$  je vrijeme  $k$ -tog povratka u stanje  $i$  te vrijedi  $T_i^{(0)} = 0$ . Kako je  $i \in S$  povratno stanje, tada imamo  $P(T_i^{(k)} < \infty \mid X_0 = i) = 1$ .

Za  $n \in \mathbb{N}$  definiramo broj izleta Markovljevog lanca iz stanja  $i$  koji cijeli stanu u interval  $[0, n]$ :

$$B(n) = \max\{k \in \mathbb{N}_0; T_i^{(k)} \leq n\},$$

za kojeg primijetimo da vrijedi

$$T_i^{(B(n))} \leq n < T_i^{(B(n)+1)}, n \in \mathbb{N}. \quad (3.3)$$

Pokažimo najprije da je

$$P \left( \left( \lim_{n \rightarrow \infty} \frac{B(n)}{n} = \frac{1}{E_i[T_i^{(1)}]} \right) \middle| X_0 = i \right) = 1.$$

Označimo s  $\alpha_n = (T_i^{(n)} - T_i^{(n-1)})$ ,  $n \in \mathbb{N}$ , duljinu trajanja  $n$ -tog izleta lanca iz stanja  $i \in S$ . Budući da su regenerativni ciklusi nezavisni i jednako distribuirani slučajni vektori,  $(\alpha_n, n \in \mathbb{N})$  je n. j. d. niz s očekivanjem  $E[\alpha_1 \mid X_0 = i]$ . Prema jakom zakonu velikih brojeva vrijedi

$$\lim_{n \rightarrow \infty} \frac{T_i^{(n)}}{n} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \alpha_k = E[\alpha_1 \mid X_0 = i] = E_i[T_i^{(1)} - T_i^{(0)}] = E_i[T_i^{(1)}] \text{ } P_i\text{-g. s.}$$

Budući da se radi o povratnom Markovljevom lancu,  $B(n) \rightarrow \infty$  za  $n \rightarrow \infty$  te slijedi:

$$\lim_{n \rightarrow \infty} \frac{T_i^{(B(n))}}{B(n)} = \lim_{n \rightarrow \infty} \frac{T_i^{(B(n)+1)}}{B(n)+1} = E_i[T_i^{(1)}] \text{ } P_i\text{-g. s.}$$

Primijetimo da je  $\lim_{n \rightarrow \infty} \frac{B(n)+1}{B(n)} = 1$  pa iz 3.3 dijeljenjem s  $B(n)$  dobivamo

$$\frac{T_i^{(B(n))}}{B(n)} \leq \frac{n}{B(n)} < \frac{T_i^{(B(n)+1)}}{B(n)+1} \cdot \frac{B(n)+1}{B(n)}$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{n}{B(n)} = E_i[T_i^{(1)}] \text{ } P_i\text{-g. s.}$$

Definirajmo slučajne varijable

$$\eta_k = \sum_{m=T_i^{(k-1)}}^{T_i^{(k)}-1} f(X_m). \quad (3.4)$$

Tada su regenerativni ciklusi  $\xi_k = (X_{T_i^{(k-1)}}, X_{T_i^{(k-1)}+1}, \dots, X_{T_i^{(k)}-1}), k \in \mathbb{N}$ , nezavisni i jednako distribuirani slučajni vektori s vrijednostima u  $E = \bigcup_{l=1}^{\infty} S^l$ . Definirajmo funkciju  $F : E \rightarrow [0, \infty)$  s

$$F(s) = \sum_{j=1}^l f(s_j), \quad s = (s_1, \dots, s_l) \in S^l.$$

Uočimo da je  $\eta_k = F(\xi_k), k \in \mathbb{N}$ , transformacija regenerativnog ciklusa. Kada  $F$  primijenimo na svaki regenerativni ciklus dobivamo n. j. d. niz  $\eta = (\eta_k, k \in \mathbb{N})$  s očekivanjem  $E[\eta_k | X_0 = i]$  te prema jakom zakonu velikih brojeva vrijedi

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{j=1}^k \eta_j = E[\eta_1 | X_0 = i] = E_i \left[ \sum_{m=0}^{T_i^{(1)}-1} f(X_m) \right], \text{ } P_i\text{-g. s.}$$

Zbog nenegativnosti funkcije  $f$ , nejednakosti 3.3 te definicije 3.4 za  $k \in \mathbb{N}$  slijedi

$$\begin{aligned} \sum_{k=1}^{B(n)} \eta_k &= \sum_{k=1}^{B(n)} \sum_{m=T_i^{(k-1)}}^{T_i^{(k)}-1} f(X_m) \\ &= \sum_{m=0}^{T_i^{(B(n))}-1} f(X_m) \\ &\leq \sum_{m=0}^{n-1} f(X_m) \\ &\leq \sum_{m=0}^{T_i^{(B(n)+1)}-1} f(X_m) \end{aligned}$$

$$\begin{aligned}
&= \sum_{K=1}^{T_i^{(B(n)+1)}} \sum_{m=T_i^{(k-1)}}^{T_i^{(k)}-1} f(X_m) \\
&= \sum_{k=1}^{B(n)+1} \eta_k.
\end{aligned}$$

Nadalje, zbog

$$1) \frac{B(n)}{n} \rightarrow \frac{1}{E_i[T_i^{(1)}]}, \quad n \rightarrow \infty, \quad P_i\text{-g. s.}$$

$$2) \frac{1}{k} \sum_{j=1}^k \eta_j \rightarrow E_i[\eta_1] = E_i \left[ \sum_{m=0}^{T_i^{(1)}-1} f(X_m) \right], \quad k \rightarrow \infty, \quad P_i\text{-g. s.}$$

slijedi

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^{B(n)} \eta_k = \lim_{n \rightarrow \infty} \frac{B(n)}{n} \cdot \frac{1}{B(n)} \sum_{k=1}^{B(n)} \eta_k = \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{m=0}^{T_i^{(1)}-1} f(X_m) \right], \quad P_i\text{-g. s.}$$

te je analogno

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^{B(n)+1} \eta_k = \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{m=0}^{T_i^{(1)}-1} f(X_m) \right], \quad P_i\text{-g. s.}$$

Iz zadnje dvije relacije te iz nejednakosti  $\sum_{k=1}^{B(n)} \eta_k \leq \sum_{k=1}^{B(n)+1} \eta_k$  slijedi

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{m=0}^{n-1} f(X_m) &= \frac{1}{E_i[T_i^{(1)}]} E_i \left[ \sum_{m=0}^{T_i^{(1)}-1} f(X_m) \right] \\
&\stackrel{3.2}{=} \pi(f) \\
&= \sum_{j \in S} f(j) \pi_j, \quad P_i\text{-g. s.}
\end{aligned}$$

Dakle, vrijedi

$$P \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{m=0}^{n-1} f(X_m) = \sum_{j \in S} f(j) \pi_j \right) = 1,$$

pa je dokaz ovime završen za nenegativne funkcije. Ako imamo proizvoljnu ograničenu funkciju,  $f$  rastavimo na pozitivni i negativni dio ( $f = f^+ - f^-$ ) te u tom kontekstu dokaz provodimo analogno kao u slučaju nenegativne funkcije za  $f^+$  i  $f^-$  pa u tom slučaju imamo  $\pi(f) = \sum_{j \in S} f(j) \pi_j = \pi(f^+) - \pi(f^-)$ .

□

**Korolar 1.**

Neka je  $f : S \rightarrow \mathbb{R}$  ograničena funkcija. Tada za svaku početnu distribuciju Markovljevog lanca  $(X_n, n \in \mathbb{N}_0)$  vrijedi

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} E[f(X_k)] = \sum_{j \in S} f(j) \pi_j = \pi(f).$$

Specijalno, za početno stanje  $i$  Markovljevog lanca te funkciju  $f = I_{\{j\}}$  imamo

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} p_{ij}^{(k)} = \pi_j.$$

Također jedno od važnih svojstava Markovljevog lanca, koje ćemo koristiti u nastavku, jest detaljna uravnoteženost (engl. *detailed balance*).

**Definicija 15.**

Neka je  $(X_n, n \in \mathbb{N}_0)$  Markovljev lanac sa skupom stanja  $S$  i stacionarnom distribucijom  $\pi$ . Za lanac kažemo da je **reverzibilan** s obzirom na  $\pi$  ako vrijedi

$$\pi_i p_{ij} = \pi_j p_{ji}, \quad \forall i, j \in S. \quad (3.5)$$

Jednadžba 3.5 zove se **uvjet detaljne uravnoteženosti**.

Primijetimo da je svojstvo 3.5 jače od uvjeta da je  $\pi$  stacionarna distribucija Markovljevog lanca o čemu nam govori sljedeći teorem.

**Teorem 5.**

Neka je  $(X_n, n \in \mathbb{N}_0)$  ergodski Markovljev lanac sa skupom stanja  $S$  i matricom 1-koračnih prijelaznih vjerojatnosti  $\Pi$ . Pretpostavimo da postoji distribucija  $\pi$  takva da je  $\pi_i p_{ij} = \pi_j p_{ji}, \forall i, j \in S$ . Tada je  $\pi$  stacionarna distribucija lanca i lanac je reverzibilan s obzirom na  $\pi$ .

*Dokaz.*

Neka vrijede pretpostavke teorema, tj. neka  $\pi$  zadovoljava uvjet  $\pi_i p_{ij} = \pi_j p_{ji}, \forall i, j \in S$ . Tada je

$$\sum_{i \in S} \pi_i p_{ij} = \sum_{i \in S} \pi_j p_{ji} = \pi_j \sum_{i \in S} p_{ji} = \pi_j.$$

Dakle, slijedi  $\pi = \pi \Pi$  pa je  $\pi$  stacionarna distribucija lanca. Prema Definiciji 3.5 Markovljev lanac  $(X_n, n \in \mathbb{N}_0)$  je reverzibilan s obzirom na  $\pi$ . □

## 3.2 Metropolis-Hastings algoritam

Metropolis-Hastings algoritam jedan je od najpopularnijih MCMC algoritama, a njegovi začetnici su *Nicholas Metropolis* i *Wilfred Keith Hastings*. Nastao je 1970. godine Hastingsovim generaliziranjem algoritma kojeg je 1953. godine razvio Metropolis. Konkretno, nastao je pokušajima matematičara da integriraju vrlo složene izraze, koji nisu rješivi analitički ili primjenom metoda numeričke integracije. Međutim, zbog svojih mogućnosti, interes za ovim algoritmom raste u područjima u kojima je potrebno simulirati uzimanje uzoraka iz proizvoljne vjerojatnosne distribucije.

### 3.2.1 Opis algoritma

Osnovna ideja metode je generiranje niza vrijednosti uzorka, tj. Markovljevog lanca sa skupom stanja  $S$  tako da njegova stacionarna distribucija dobro reprezentira zadanu distribuciju  $\pi$ , koju nazivamo **ciljna distribucija**. Kod standardne analize Markovljevih lanaca poznata nam je funkcija prijelaznih vjerojatnosti, a nepoznata je stacionarna distribucija, dok je u MCMC pristupu poznata stacionarna distribucija, a funkcija prijelaznih vjerojatnosti se definira tako da se stacionarna distribucija postigne (u konačnom vremenu). Rezultirajući Markovljev lanac je ergodski pa je prema Teoremu 3 stacionarna distribucija ujedno i granična distribucija. Dakle, uz dovoljno veliki broj iteracija, stacionarna distribucija lanca postaje ciljna distribucija. Algoritam započinjemo s početnim stanjem  $x^{(0)}$  te iterativno nadograđujemo Markovljev lanac, tako da u svakoj iteraciji odabiremo kandidata za sljedeću vrijednost uzorka na temelju trenutne vrijednosti uzorka. Stoga, definiramo funkciju prijedloga stanja  $x'$  izrazom  $q(x, x') = q(x'|x)$ . Dakle, uvjetna vjerojatnost  $q(x'|x)$  predstavlja vjerojatnost prelaska Markovljevog lanca iz stanja  $x$  u stanje  $x'$ . Stanje  $x'$  možemo odbiti ili prihvatiti kako je opisano u Algoritmu 1.

---

#### Algoritam 1 Metropolis-Hastings

---

- 1: Odaberi  $x^{(0)}$  tako da vrijedi  $\pi(x^{(0)}) > 0$  i broj iteracija  $T$ ;
- 2: **za**  $t = 0, 1, 2, \dots, T$  **radi**
- 3:  $x = x^{(t)}$ ;
- 4:  $x' \sim q(x'|x)$ ;
- 5: Izračunaj vjerojatnost  $\alpha(x'|x)$  po pravilu

$$\alpha(x'|x) = \min \left\{ \frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)}, 1 \right\};$$

- 6:  $u \sim U(0, 1)$ ;

7:

$$\text{Definiraj } x^{(t+1)} = \begin{cases} x', & \text{ako je } u < \alpha(x'|x) \\ x, & \text{ako je } u \geq \alpha(x'|x) \end{cases}$$


---

Ako je kandidat odbijen, vrijednost kandidata se odbacuje te se trenutna vrijednost ponovno koristi u sljedećoj iteraciji, dok se u situaciji kad je kandidat prihva-

ćen, koristi vrijednost kandidata. Vrijednost  $\alpha(x'|x)$  iz koraka 5 zovemo **Metropolis–Hastingsova vjerojatnost prihvatanja** i ona govori kolika je vjerojatnost da prihvatimo novog kandidata. Izraz

$$\frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)} \quad (3.6)$$

zovemo **Hastingsov omjer**. Omjer nije definiran ako je  $\pi(x) = 0$ , stoga na početku algoritma zahtijevamo da vrijedi  $\pi(x^{(0)}) > 0$ . Također, trebamo imati na umu da kandidat  $x'$  treba zadovoljavati uvjet  $q(x'|x) > 0$  s vjerojatnošću 1 kako bi omjer bio dobro definiran. 1–koračna prijelazna vjerojatnost iz stanja  $x$  u stanje  $x'$  je tada dana izrazom  $p_{xx'} = q(x'|x) \cdot \alpha(x'|x)$ .

Uočimo:

- (i) ako je  $\alpha(x'|x) = 1$  tada je  $\frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)} \geq 1$ , tj.  $\frac{\pi(x) \cdot q(x'|x)}{\pi(x') \cdot q(x|x')} < 1$  pa je  $\alpha(x|x') < 1$ ,
- (ii) ako je  $\alpha(x'|x) < 1$  tada je  $\frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)} < 1$ , tj.  $\frac{\pi(x) \cdot q(x'|x)}{\pi(x') \cdot q(x|x')} > 1$  pa je  $\alpha(x|x') = 1$ .

Pretpostavimo da vrijedi  $\alpha(x'|x) < 1$  i  $\alpha(x|x') = 1$ . Pokažimo da je zadovoljen uvjet detaljne uravnoteženosti, tj.

$$\begin{aligned} \alpha(x'|x) &= \frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)} \\ \iff \pi(x) \cdot q(x'|x) \cdot \alpha(x'|x) &= \pi(x') \cdot q(x|x') \\ \iff \pi(x) \cdot q(x'|x) \cdot \alpha(x'|x) &= \pi(x') \cdot q(x|x') \cdot \alpha(x|x') \\ \iff \pi(x) \cdot p_{xx'} &= \pi(x') \cdot p_{x'x}. \end{aligned}$$

Analogno se pokaže i za drugi slučaj. Dakle,  $\pi$  je stacionarna distribucija Markovljevog lanca.

Ako je  $q$  simetrična, tj. ako vrijedi  $q(x'|x) = q(x|x')$ , tada je izraz 3.6 jednak  $\frac{\pi(x')}{\pi(x)}$  te se postupak svodi na Metropolis algoritam.

### 3.2.2 Specijalni slučajevi Metropolis–Hastings algoritma

Postoje različite varijante ovog algoritma, a mi ćemo spomenuti dvije koje se u praksi najčešće koriste.

Ako je funkcija prijedloga  $q$  u algoritmu nezavisna o trenutnom stanju u Markovljevom lancu, onda govorimo o **nezavisnom Metropolis–Hastings** algoritmu. Dakle, funkcija prijedloga u ovom slučaju glasi  $q(x, x') = q(x')$ . Najčešće se u praksi za funkciju prijedloga koristi normalna (Gaussova) distribucija s očekivanjem  $x$ , tako da je vjerojatnije da će točke bliže  $x$  biti sljedeće posjećene. Iako su vrijednosti, koje dobijemo ovim algoritmom, realizacije nezavisnih slučajnih varijabli, vjerojatnost prihvatanja novog kandidata ovisi o prethodnom stanju pa

**Algoritam 2** Nezavisni Metropolis–Hastings

- 
- 1: Odaberi  $x^{(0)}$  tako da vrijedi  $\pi(x^{(0)}) > 0$  i broj iteracija  $T$ ;
  - 2: **za**  $t = 0, 1, 2, \dots, T$  **radi**
  - 3:  $x = x^{(t)}$ ;
  - 4:  $x' \sim q(x')$ ;
  - 5: Izračunaj vjerojatnost  $\alpha(x'|x)$  po pravilu

$$\alpha(x'|x) = \min \left\{ \frac{\pi(x') \cdot q(x)}{\pi(x) \cdot q(x')}, 1 \right\};$$

- 6:  $u \sim U(0, 1)$ ;

7:

$$\text{Definiraj } x^{(t+1)} = \begin{cases} x', & \text{ako je } u < \alpha(x'|x) \\ x, & \text{ako je } u \geq \alpha(x'|x) \end{cases}$$


---

uzorak nije nezavisan. Ako su prijedlozi oblika  $x' = x + \epsilon_t$ , gdje je  $\{\epsilon_t\}$  uzorak od konačno mnogo slučajno odabranih realizacija neke distribucije (najčešće se uzima simetrična distribucija), onda se radi o **Metropolis–Hastings algoritmu sa slučajnom šetnjom**. Metoda se zasniva na lokalnom pretraživanju okoline trenutne vrijednosti Markovljevog lanca. U tom slučaju funkcija prijedloga je oblika:

$$q(x, x') = q(x'|x) = q(x' - x) = q(\epsilon_t),$$

$$q(x', x) = q(x|x') = q(x - x') = q(-\epsilon_t).$$

**Algoritam 3** Metropolis–Hastings sa slučajnom šetnjom

- 
- 1: Odaberi  $x^{(0)}$  tako da vrijedi  $\pi(x^{(0)}) > 0$  i broj iteracija  $T$ ;
  - 2: **za**  $t = 0, 1, 2, \dots, T$  **radi**
  - 3:  $x = x^{(t)}$ ;
  - 4:  $x' \sim q(x'|x)$ ;
  - 5: Izračunaj vjerojatnost  $\alpha(x'|x)$  po pravilu

$$\begin{aligned} \alpha(x'|x) &= \min \left\{ \frac{\pi(x') \cdot q(x|x')}{\pi(x) \cdot q(x'|x)}, 1 \right\} \\ &= \min \left\{ \frac{\pi(x') \cdot q(-\epsilon_t)}{\pi(x) \cdot q(\epsilon_t)}, 1 \right\}; \end{aligned}$$

- 6:  $u \sim U(0, 1)$ ;

7:

$$\text{Definiraj } x^{(t+1)} = \begin{cases} x', & \text{ako je } u < \alpha(x'|x) \\ x, & \text{ako je } u \geq \alpha(x'|x) \end{cases}$$


---

Ako je  $\epsilon_t$  simetrična distribucija onda vrijedi  $q(\epsilon_t) = q(-\epsilon_t)$  pa je u tom slučaju vjerojatnost prihvatanja  $\alpha(x'|x) = \min \left\{ \frac{\pi(x+\epsilon_t)}{\pi(x)}, 1 \right\}$ .

Promotrimo sada sljedeće slučajeve prilikom odabira kandidata:

- (i) ako su pomaci  $\epsilon_t$  minorni (tj. varijanca od  $\epsilon_t$  je mala), tada je omjer  $\frac{\pi(x+\epsilon_t)}{\pi(x)}$  blizu 1 pa se kandidati skoro uvijek prihvaćaju;
- (ii) ako su pomaci  $\epsilon_t$  veliki (tj. varijanca od  $\epsilon_t$  je prevelika), tada je omjer  $\frac{\pi(x+\epsilon_t)}{\pi(x)}$  blizu 0 pa se kandidati skoro uvijek odbijaju.

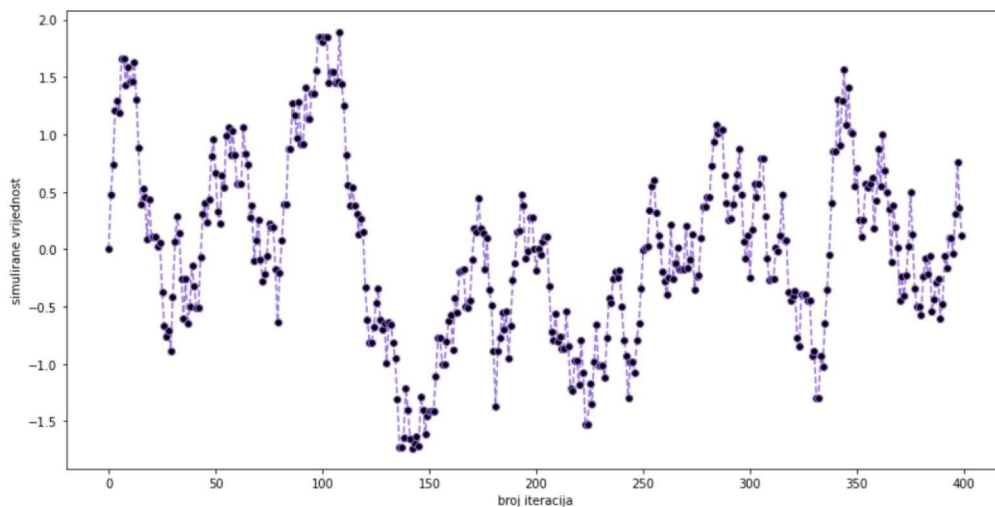
Odabirom adekvatne varijance od  $\epsilon_t$  bave se **adaptivne Metropolis–Hastings metode**, a detaljnije o njima možete pronaći u [9].

Pokažimo na jednostavnom primjeru kako se Metropolis-Hastings sa slučajnom šetnjom može koristiti za uzorkovanje iz standardne normalne distribucije.

### Primjer 2.

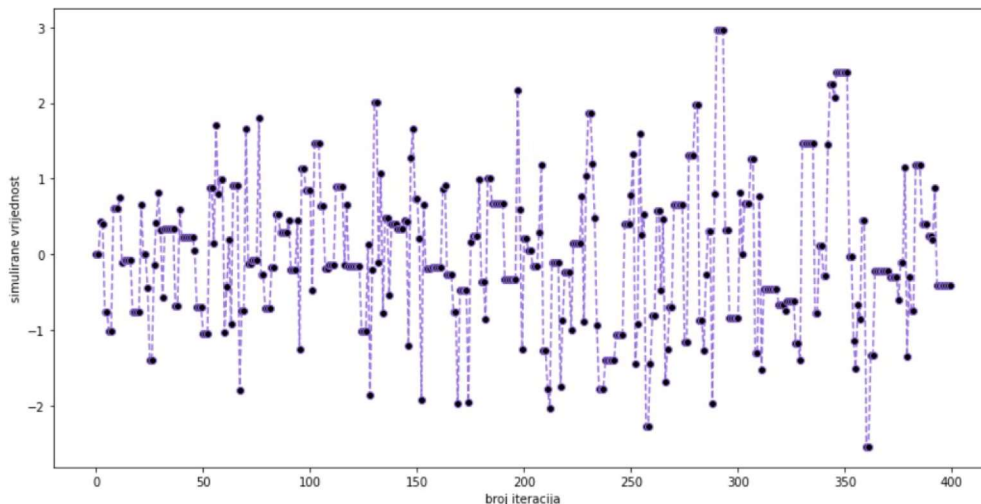
Neka je distribucija slučajne varijable  $\epsilon_t$  dana s  $\epsilon_t \sim U(-\delta, \delta)$ , gdje je  $\delta > 0$  proizvoljan. Promotrit ćemo slučajeve kada je  $\delta$  jednako 0.5 i 3. Za dani  $\epsilon_t$  definiramo  $x' = x + \epsilon_t$  kao što smo opisali u algoritmu 3. Dakle, od stanja u kojem se trenutno nalazimo se pomaknemo za jako mali  $\epsilon_t$  te na taj način istražujemo lokalnu okolinu trenutnog stanja. Budući da želimo simulirati standardnu normalnu distribuciju, tada je ona i ciljna distribucija, tj.  $\pi \sim \mathcal{N}(0, 1)$ . Neka je broj iteracija algoritma jednak 400 te bez smanjenja općenitosti svaku simulaciju započinjemo iz točke 0.

Na dijagramima 3.1 i 3.2 su prikazane simulacije Markovljevog lanca za  $\delta = 0.5$  i  $\delta = 3$ .



Slika 3.1: Simulirana trajektorija Markovljevog lanca za  $\delta = 0.5$

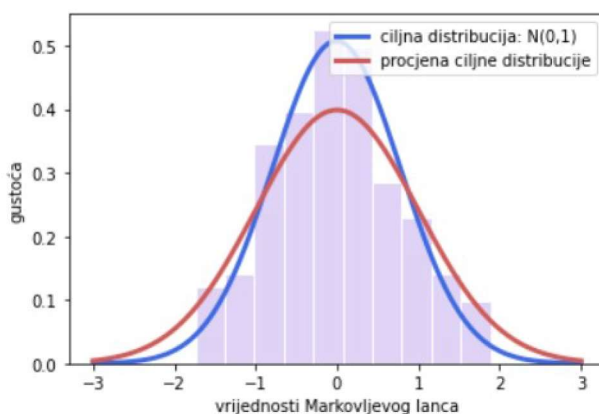




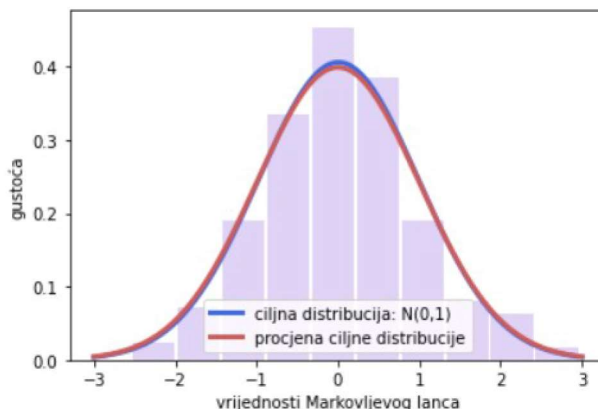
Slika 3.2: Simulirana trajektorija Markovljevog lanca za  $\delta = 3$

Na dijagramu 3.2 vidimo da postoji niz iteracija u kojima nema pomaka, tj. predloženi kandidati su odbijeni, dok je na dijagramu 3.1 bilo više pomaka. Možemo zaključiti da u slučaju kada je  $\delta$  velik, kandidati koje predlažemo mogu biti previše udaljeni od stanja u kojem se Markovljev lanac trenutno nalazi pa je stoga veća vjerojatnost da će kandidat biti odbijen. U slučaju malih vrijednosti za  $\delta$ , vrijednosti  $\pi(x')$  i  $\pi(x)$  su približno jednake pa je vjerojatnost prihvatanja novog kandidata uvijek približno jedan.

Na dijagramima 3.3 i 3.4 su prikazani histogrami simuliranih vrijednosti Markovljevog lanca za  $\delta = 0.5$  i  $\delta = 3$ . Očekivanje simulirane distribucije s dijagrama 3.3 iznosi  $-0.007$ , dok je varijanca jednaka  $0.786$ . U slučaju kada je  $\delta = 3$  (dijagram 3.4), očekivanje je  $0.007$ , dok varijanca iznosi  $0.983$ .



Slika 3.3: Histogram simuliranih vrijednosti Markovljevog lanca za  $\delta = 0.5$



Slika 3.4: Histogram simuliranih vrijednosti Markovljevog lanca za  $\delta = 3$

Zaključno, neovisno o vrijednosti  $\delta$ , jednodimenzionalne distribucije Markovljeva lanca konvergiraju prema standardnoj normalnoj distribuciji. Brzina konvergencije jednodimenzionalnih distribucija prema stacionarnoj distribuciji koja je blizu ciljne distribucije, kao i veličina skupa stanja koja posjetimo prilikom simulacija, ovise o  $\delta$ .

### 3.3 Gibbsovo uzorkovanje

Kao i druge MCMC metode, Gibbsovo uzorkovanje konstruira Markovljev lanac čije jednodimenzionalne distribucije konvergiraju prema ciljnoj distribuciji. Ime je dobilo po fizičaru Josiahu Willardu Gibbsu, a algoritam su opisali braća Stuart i Donald Geman 1984. godine. Radi se zapravo o specifičnom slučaju Metropolis-Hastings algoritma u kojem se prijedlozi uvijek prihvaćaju. Za razliku od prethodnog algoritma, Gibbsovo uzorkovanje primjenjivo je jedino u višedimenzionalnim problemima kada zajednička distribucija slučajnih varijabli nije eksplicitno poznata ili je teško iz nje izravno uzorkovati, ali su nam poznate uvjetne distribucije svake slučajne varijable. Primjene ovog algoritma mogu se pronaći u genetici (npr. zaključivanje o mješovitom nasljeđivanju u životinjskoj populaciji), strojnom učenju (npr. otklanjanje šuma sa slike), medicini, bioinformatici, fizici, statistici i sl.

---

#### Algoritam 4 Gibbsovo uzorkovanje

---

- 1: Inicijaliziraj  $x^{(0)} = (x_1^0, \dots, x_n^0)$  i broj iteracija  $T$ ;
  - 2: **za**  $t = 0, 1, 2, \dots, T$  **radi**
  - 3:     **za**  $i = 1, 2, \dots, n$  **radi**
  - 4:          $x_i^t \sim \pi(x_i^t | x_1^t, \dots, x_{i-1}^t, x_{i+1}^{t-1}, \dots, x_n^{t-1})$ ;
- 

Dakle, iz Algoritma 4 možemo primijetiti da ažuriranjem  $x_i^t$ , u sljedećem koraku koristimo njegovu vrijednost za uzorkovanje  $x_{i+1}^t$  vrijednosti. U ovom slučaju, funkcija prelaska je oblika

$$q(x_i', \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i}) = \pi(x_i' | \mathbf{x}_{-i}),$$

gdje  $\mathbf{x}_{-i}$  označava vektor  $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ .

Pokažimo sada da se radi o Metropolis–Hastings algoritmu, u kojem se kandidati uvijek prihvaćaju. Konkretno, trebamo pokazati da je vjerojatnost prihvaćanja  $\alpha$  iz Algoritma 1 uvijek jednaka jedan:

$$\begin{aligned} \alpha(x'_i, \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i}) &= \min \left\{ \frac{\pi(x'_i, \mathbf{x}_{-i}) \cdot q(x_i, \mathbf{x}_{-i} | x'_i, \mathbf{x}_{-i})}{\pi(x_i, \mathbf{x}_{-i}) \cdot q(x'_i, \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i})}, 1 \right\} \\ &= \min \left\{ \frac{\pi(x'_i, \mathbf{x}_{-i}) \cdot \pi(x_i | \mathbf{x}_{-i})}{\pi(x_i, \mathbf{x}_{-i}) \cdot \pi(x'_i | \mathbf{x}_{-i})}, 1 \right\} \\ &= \min \left\{ \frac{\pi(x'_i | \mathbf{x}_{-i}) \cdot \pi(\mathbf{x}_{-i}) \cdot \pi(x_i | \mathbf{x}_{-i})}{\pi(x_i | \mathbf{x}_{-i}) \cdot \pi(\mathbf{x}_{-i}) \cdot \pi(x'_i | \mathbf{x}_{-i})}, 1 \right\} \\ &= \min\{1, 1\} \\ &= 1. \end{aligned}$$

Pogledajmo na jednostavnom primjeru kako funkcionira Gibbsovo uzorkovanje.

### Primjer 3.

Neka su  $X$  i  $Y$  slučajne varijable s Bernoullijevom distribucijom te neka je dana zajednička distribucija  $P(X = x, Y = y) = p_{X,Y}(x, y)$  tablicom:

$X \setminus Y$	0	1	
0	0.4	0.2	0.6
1	0.1	0.3	0.4
	0.5	0.5	

Tada je uvjetna distribucija od  $X$  uz uvjet  $\{Y = y\}$ :

$$P(X = 0 | Y = 0) = \frac{P(X = 0, Y = 0)}{P(Y = 0)} = \frac{0.4}{0.5} = 0.8,$$

$$P(X = 0 | Y = 1) = \frac{P(X = 0, Y = 1)}{P(Y = 1)} = \frac{0.2}{0.5} = 0.4,$$

$$P(X = 1 | Y = 0) = \frac{P(X = 1, Y = 0)}{P(Y = 0)} = \frac{0.1}{0.5} = 0.2,$$

$$P(X = 1 | Y = 1) = \frac{P(X = 1, Y = 1)}{P(Y = 1)} = \frac{0.3}{0.5} = 0.6.$$

Analogno dobivamo uvjetnu distribuciju od  $Y$  uz uvjet  $\{X = x\}$ :

$$P(Y = 0 | X = 0) = \frac{P(Y = 0, X = 0)}{P(X = 0)} = \frac{0.4}{0.6} = 0.67,$$

$$P(Y = 0 | X = 1) = \frac{P(Y = 0, X = 1)}{P(X = 1)} = \frac{0.1}{0.4} = 0.25,$$

$$P(Y = 1|X = 0) = \frac{P(Y = 1, X = 0)}{P(X = 0)} = \frac{0.2}{0.6} = 0.33,$$

$$P(Y = 1|X = 1) = \frac{P(Y = 1, X = 1)}{P(X = 1)} = \frac{0.3}{0.4} = 0.75.$$

Pretpostavimo da krenemo s nekim vrijednostima od  $X$  i  $Y$  i ponavljamo sljedeća dva koraka:

1. generiramo vrijednost od  $X$  prema  $P(X|Y = y)$ , gdje je  $y$  trenutna vrijednost od  $Y$ ;
2. generiramo vrijednost od  $Y$  prema  $P(Y|X = x)$ , gdje je  $x$  vrijednost generirana u koraku 1.

Bez smanjenja općenitosti, neka je  $(0, 0)$  početno stanje. Ako pokrenemo gore opisani algoritam za broj iteracija  $n=10000$ , dobivamo Markovljev lanac čija je stacionarna distribucija dana tablicom:

$X \setminus Y$	0	1	
0	0.4048	0.1989	0.6037
1	0.1009	0.2954	0.3963
	0.5057	0.4943	

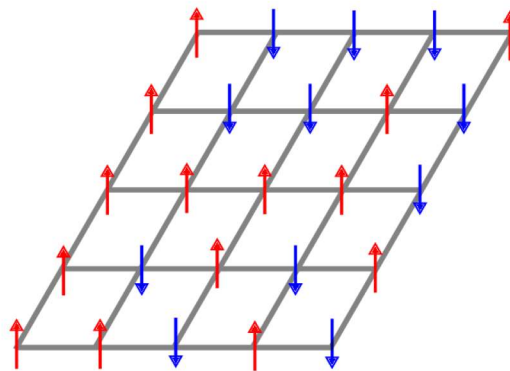
Uočimo da stacionarna distribucija dobro aproksimira ciljnu distribuciju s početka primjera.

## 4 | Primjena MCMC metode u obradi slika

Spomenuli smo da se MCMC metoda koristi u mnogim problemima iz stvarnog svijeta. U ovom poglavlju ćemo obraditi primjenu MCMC metode u obradi slika, a riječ je o redukciji šuma binarne slike. Šum obično vežemo uz nešto nepoželjno, a u terminima slike možemo ga definirati kao varijaciju svjetline ili boje. U svrhu redukcije šuma koristit ćemo Metropolis–Hastingsov algoritam, kojega smo ranije opisali. Za potrebe ovog rada, pretpostavit ćemo da je slika koju želimo simulirati generirana korištenjem Isingovog modela, stoga ćemo najprije ukratko opisati njegovu strukturu.

### 4.1 Isingov model

Isingov model, nazvan po fizičaru *Ernstu Isingu*, je matematički model koji se koristi u statističkoj mehanici. Model je predstavljen pravilnom rešetkom atomskih spinova koji mogu biti u jednom od dva stanja ( $-1$  ili  $+1$ ). Svaki spin međusobno komunicira sa susjednim spinovima. Ako se susjedni spinovi slažu (tj. imaju istu vrijednost) tada imaju nižu energiju od onih koji se ne slažu te sustav teži najnižoj energiji.



Slika 4.1: Prikaz 2D Isingovog modela. Slika preuzeta iz [15].

#### Definicija 16.

Neka je  $\Lambda$  skup elemenata, od kojih svaki element ima skup susjeda, čineći tako kvadratnu rešetku. Za svaki element  $k \in \Lambda$  postoji diskretna varijabla  $\sigma_k$  tako da je  $\sigma_k \in \{-1, +1\}$ .

Dodjelu vrijednosti spina svakom elementu unutar rešetke označavamo sa  $\sigma = \{\sigma_k\}_{k \in \Lambda}$  i zovemo ju **konfiguracija spina**.

Za svaka dva susjedna elementa  $i, j \in \Lambda$  postoji interakcija  $J_{ij}$ . Također, element  $j \in \Lambda$  ima vanjsko magnetno polje  $h_j$  koje je u interakciji s njim. Energija konfiguracije  $\sigma$  je dana Hamiltonovom funkcijom

$$H(\sigma) = - \sum_{\langle ij \rangle} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j, \quad (4.1)$$

gdje oznaka  $\langle ij \rangle$  predstavlja da su elementi  $i$  i  $j$  susjedi.

Vjerojatnost konfiguracije je zadana višedimenzionalnom Boltzmannovom distribucijom

$$\pi_\beta(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z_\beta}, \quad (4.2)$$

gdje je  $\beta \geq 0$ , a  $Z_\beta$  zovemo konstanta normalizacije i dana je izrazom

$$Z_\beta = \sum_{\sigma} e^{-\beta H(\sigma)}.$$

U praksi se često Isingov model koristi bez korištenja vanjskog polja u interakciji s rešetkom, tj.  $h_j = 0, \forall j \in \Lambda$ . Također, možemo pretpostaviti da je  $J_{ij}$  jednak za svaki element  $i$  u tom slučaju ga označavamo s  $J$ . Tada izraz 4.1 možemo pojednostavniti:

$$H(\sigma) = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j. \quad (4.3)$$

Ukoliko postoji previše stanja u sustavu kojeg promatramo, tada Isingov model možemo teško procijeniti. Promotrimo Isingov sustav koji ima  $L = |\Lambda|$  elemenata unutar rešetke te svaki element može imati jednu od dvije moguće vrijednosti (-1 ili +1). Tada je ukupni broj mogućih stanja jednak  $2^L$  pa zbog toga Isingov model možemo simulirati korištenjem MCMC metode.

Budući da imamo ukupno  $L$  mogućih elemenata, vjerojatnost odabira elementa u rešetki jednaka je  $q(x'|x) = \frac{1}{L}$ . Zbog detaljne uravnoteženosti, vjerojatnost prihvaćanja novog stanja jednaka je

$$\alpha(x'|x) = \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} = \frac{\pi(x')\frac{1}{L}}{\pi(x)\frac{1}{L}} = \frac{\pi(x')}{\pi(x)} = \frac{\frac{1}{Z_\beta} e^{-\beta H(x')}}{\frac{1}{Z_\beta} e^{-\beta H(x)}} = e^{-\beta(H(x')-H(x))}. \quad (4.4)$$

Detaljnije o Isingovom modelu možete pronaći u [16].

## 4.2 Redukcija šuma na slici pomoću MCMC metode

Pretpostavimo da imamo binarnu sliku sa šumom  $X$ . Binarna slika se može promatrati kao kvadratna rešetka, gdje svaki piksel predstavlja jedan element unutar

rešetke. Ovisno o boji piksela, svaki piksel može poprimiti vrijednost 1 (bijela boja) ili -1 (crna boja). Šum na binarnoj slici postiže se poprimanjem suprotne vrijednosti piksela od vrijednosti piksela na originalnoj slici. Originalna slika nam je nepoznata i nju ćemo označiti s  $Y$ . Cilj nam je od slike  $X$  dobiti sliku  $Y$ . Pretpostavimo da je  $Y$  generirana Isingovim modelom pa želimo pomoću Metropolis–Hastings algoritma simulirati Boltzmanovu distribuciju 4.2. U nastavku dajemo algoritam za smanjenje šuma na slici.

---

**Algoritam 5** Redukcija šuma na slici pomoću Metropolis–Hastings algoritma
 

---

- 1: Incijaliziraj  $Y^{(0)} \leftarrow X$ , odaberi parametre  $J, \beta \geq 0$  i broj iteracija  $T$ ;
- 2: **za**  $t = 0, 1, 2, \dots, T$  **radi**
- 3: Na slučajan način odaberi piksel  $Y_{ij}^{(t)}$ ;
- 4: Predloži kandidata  $Y'_{ij} = -Y_{ij}^{(t)}$ ;
- 5: Izračunaj vjerojatnost prihvatanja

$$\alpha(Y'_{ij}|Y_{ij}^{(t)}) = \min \left\{ e^{-2\beta J \sum_{(i,j) \sim (k,l)} Y_{ij}^{(t)} Y_{kl}^{(t)}}, 1 \right\};$$

- 6: Generiraj  $u \sim U(0, 1)$ ;

7:

$$\text{Definiraj } Y_{ij}^{(t+1)} = \begin{cases} Y'_{ij}, & \text{ako je } u < \alpha(Y'_{ij}|Y_{ij}^{(t)}) \\ Y_{ij}^{(t)}, & \text{ako je } u \geq \alpha(Y'_{ij}|Y_{ij}^{(t)}) \end{cases}$$


---

Zbog koraka 4 algoritma 5 i jednakosti 4.4, vjerojatnost prihvatanja iz koraka 5 je dana s:

$$\begin{aligned} \alpha(Y'_{ij}|Y_{ij}^{(t)}) &= \min \left\{ e^{-\beta(H(Y'_{ij}) - H(Y_{ij}^{(t)}))}, 1 \right\} \\ &= \min \left\{ e^{-\beta \left( -J \sum_{(i,j) \sim (k,l)} Y'_{ij} Y_{kl}^{(t)} + J \sum_{(i,j) \sim (k,l)} Y_{ij}^{(t)} Y_{kl}^{(t)} \right)}, 1 \right\} \\ &= \min \left\{ e^{-\beta \left( J \sum_{(i,j) \sim (k,l)} Y_{ij}^{(t)} Y_{kl}^{(t)} + J \sum_{(i,j) \sim (k,l)} Y_{ij}^{(t)} Y_{kl}^{(t)} \right)}, 1 \right\} \\ &= \min \left\{ e^{-2\beta J \sum_{(i,j) \sim (k,l)} Y_{ij}^{(t)} Y_{kl}^{(t)}}, 1 \right\}. \end{aligned}$$

Na sljedećem grafičkom prikazu možemo vidjeti jednu iteraciju algoritma 5. Neka su zadani parametri  $\beta = 0.9$  te  $J = 1$ . Na slici (a) se nalazi primjer inicijalne slike koju imamo na ulazu algoritma. Na slici (b) je prikazana slika nakon  $(t - 1)$  iteracija algoritma. U iteraciji  $t$  (slika (c)), slučajno odaberemo jedan piksel. Neka je to piksel  $y_{24}^{(t)}$  koji ima vrijednost 1. Predložimo kandidata  $Y'_{24} = -Y_{24}^{(t)}$  te računamo vjerojatnost prihvatanja kandidata

$$\alpha(1|-1) = \min \{ e^{-2 \cdot 0.9 \cdot (-1) \cdot (1+1+1+1)}, 1 \} = \min \{ e^{7.2}, 1 \} = 1.$$

Dakle, s vjerojatnošću 1 prihvaćamo predloženog kandidata, a rezultat vidimo na slici (d).

-1	-1	1	1	1
-1	1	1	-1	1
-1	1	1	1	1
-1	-1	-1	1	-1
1	-1	-1	1	1

(a) Inicijalna slika  $Y^{(0)} = X$ 

-1	-1	1	1	1
-1	1	1	-1	1
1	1	1	1	1
-1	1	-1	1	-1
1	-1	-1	1	1

(b) Slika nakon  $(t - 1)$  iteracija ( $Y^{(t-1)}$ )

-1	-1	1	1	1
-1	1	1	-1	1
-1	1	1	1	1
-1	-1	-1	1	-1
1	-1	-1	1	1

(c) U iteraciji  $t$  slučajno odaberemo piksel (npr.  $Y_{24}^{(t)}$ ) te pronađemo njegove susjede

-1	-1	1	1	1
-1	1	1	1	1
1	1	1	1	1
-1	1	-1	1	-1
1	-1	-1	1	1

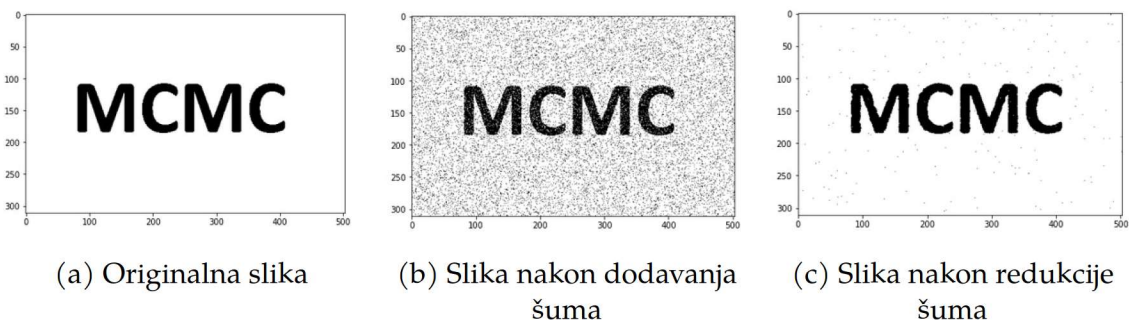
(d) Slika u iteraciji  $(t + 1)$ , nakon što smo promijenili vrijednost piksela  $Y_{24}^{(t)}$ 

Slika 4.2: Grafički prikaz jedne iteracije algoritma 5

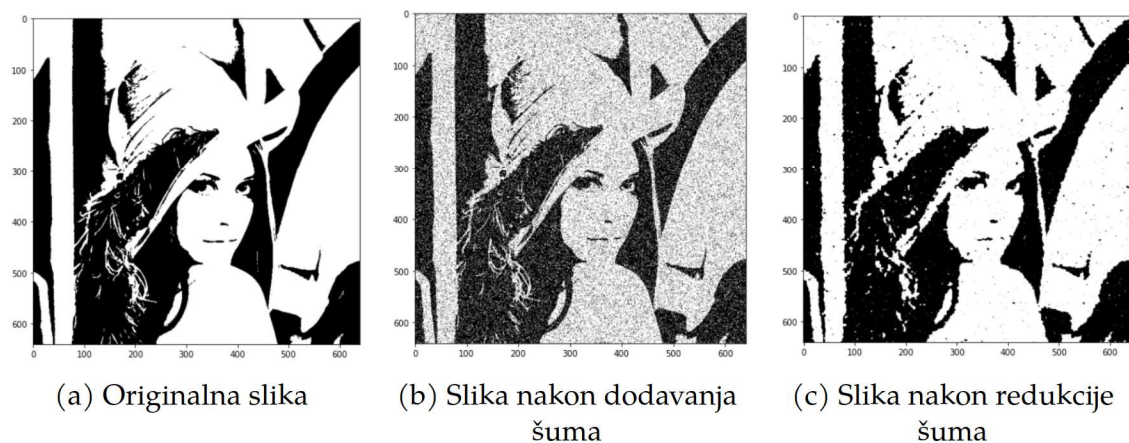
Na sljedećim grafičkim prikazama možemo vidjeti rezultate algoritma, koje smo dobili pomoću programskog jezika *Python*. Na originalnu sliku smo dodali šum te smo nakon toga pokrenuli gore opisani algoritam kako bismo ga otklonili. Slika 4.4 (*Lena*) često je korištena u obradi slika te je preuzeta s <https://tinyurl.com/ywr3u5nc>. Prilikom pokretanja algoritma koristili smo sljedeće parametre:

Slika	Postotak šuma na slici	Broj iteracija algoritma	$\beta$	$J$
4.3	10%	500000	0.9	1
4.4	15%	2500000	0.95	1
4.5	22%	1500000	1	1

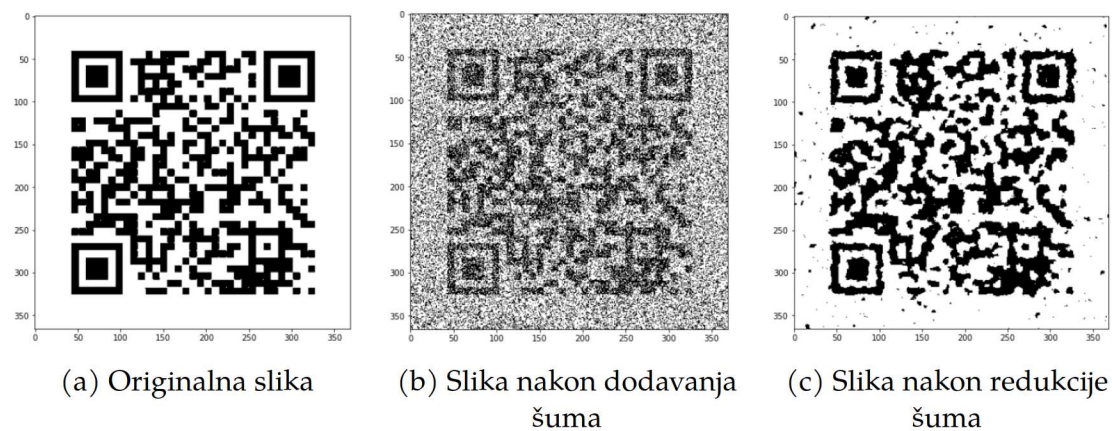




Slika 4.3: Otklanjanje šuma sa slike teksta



Slika 4.4: Otklanjanje šuma s portreta



Slika 4.5: Otklanjanje šuma s QR koda



## 5 | Kod

### Primjer 1

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 def definite_integral_show(f, x0, x1, N):
4     x = np.arange(x0, x1, 0.01)
5     y = f(x)
6     f_max = max(y)
7
8     x_rand = x0 + np.random.random(N)*(x1-x0)
9     y_rand = 0 + np.random.random(N)*f_max
10
11     ind_below = np.where(y_rand < f(x_rand))
12     ind_above = np.where(y_rand >= f(x_rand))
13
14     plt.plot(x, y, color = "black",linewidth=3)
15     pts_below = plt.scatter(x_rand[ind_below], y_rand[ind_below],
16                             color = "mediumpurple")
17     pts_above = plt.scatter(x_rand[ind_above], y_rand[ind_above],
18                             color = "mediumpurple")
19     plt.show()
20
21     plt.plot(x, y, color = "black",linewidth=3)
22     pts_below = plt.scatter(x_rand[ind_below], y_rand[ind_below],
23                             color = "mediumturquoise")
24     pts_above = plt.scatter(x_rand[ind_above], y_rand[ind_above],
25                             color = "mediumpurple")
26     plt.show()
27
28     print("Broj tocaka iznad grafa funkcije:", len(ind_above[0]))
29     print("Broj tocaka ispod grafa funkcije:", len(ind_below[0]))
30     print("Br. tocaka ispod grafa/Ukupan br. tocaka:", len(
31         ind_below[0])/N)
32     print("Povrsina pravokutnika:", f_max*(x1-x0))
33     print("Povrsina ispod grafa funkcije:", f_max*(x1-x0)*len(
34         ind_below[0])/N)
35
36 def g(x):
37     return np.sin(x)
38
39 for N in [100,1000,10000]:
40     definite_integral_show(g, 0, np.pi, N)
```

## Primjer 2

```
1 import numpy as np
2 import random
3 from scipy.stats import norm
4 import matplotlib.pyplot as plt
5 import statistics
6 import pandas as pd
7
8 delta = np.array([0.5, 3])
9 T = 400
10 X = np.zeros(T)
11
12 for i in range(len(delta)):
13     for t in range(1,T):
14         epsilon = np.random.uniform(-delta[i], delta[i])
15         y = X[t-1] + epsilon
16         alpha = np.minimum(norm.pdf(y)/norm.pdf(X[t-1]),1)
17         u = np.random.uniform(0,1)
18         if u < alpha:
19             X[t] = y
20         else:
21             X[t] = X[t-1]
22
23 mu, std = norm.fit(X)
24 print(mu)
25 print(std)
26 plt.hist(X, bins=10, rwidth=0.95,density=True, alpha=0.3, color="
    mediumpurple")
27 xmin, xmax = plt.xlim()
28 x = np.linspace(-3, 3, T)
29 p = norm.pdf(x, mu, std)
30 q = norm.pdf(x, 0, 1)
31 plt.plot(x, p, 'k', linewidth=3, color = 'royalblue', label='
    ciljna distribucija: N(0,1)')
32 plt.plot(x, q, 'k', linewidth=3, color = 'indianred', label='
    procjena ciljne distribucije')
33 plt.legend()
34 plt.xlabel("vrijednosti Markovljevog lanca")
35 plt.ylabel("gusto a")
36
37 dfDistrHist = pd.DataFrame(X)
38 dfDistrHist.plot(color="mediumpurple", legend=False)
39 plt.xlabel("broj iteracija")
40 plt.ylabel("simulirane vrijednost")
41 plt.show()
```

## Primjer 3

```
1 import numpy as np
2 import random
3 from scipy.stats import bernoulli
4
5 n = 10000
6 X = np.zeros(n)
7 Y = np.zeros(n)
8
9 res00=1
10 res01=0
11 res10=0
12 res11=0
13
14 def X_uz_uvjet_Y(y):
15     if(y==0):
16         x = bernoulli.rvs(0.2)
17     else:
18         x = bernoulli.rvs(0.6)
19     return(x)
20
21 def Y_uz_uvjet_X(x):
22     if(x==0):
23         y = bernoulli.rvs(0.33)
24     else:
25         y = bernoulli.rvs(0.75)
26     return(y)
27
28 for i in range(1,n):
29     X[i] = X_uz_uvjet_Y(Y[i-1])
30     Y[i] = Y_uz_uvjet_X(X[i])
31     if((X[i]==0) & (Y[i]==0)):
32         res00+=1
33     if((X[i]==0) & (Y[i]==1)):
34         res01+=1
35     if((X[i]==1) & (Y[i]==0)):
36         res10+=1
37     if((X[i]==1) & (Y[i]==1)):
38         res11+=1
39
40 result = np.column_stack((X,Y))
41 print(result[:10,:])
42 print(res00/n, res01/n, res10/n, res11/n)
```

## Primjena MCMC metode u obradi slika

```
1 from google.colab import drive
2 drive.mount("/content/gdrive")
3
4 from PIL import Image
5 import numpy as np
6 from matplotlib import pyplot as plt
7
8 original = Image.open("/content/gdrive/MyDrive/Diplomski/MCMC.png")
9 original = original.convert('L')
10 original = np.asarray(original)
11 original = 2*(original > 128).astype(int)-1
12
13 pi = 0.1
14 m,n = original.shape
15 flip = np.random.rand(m,n) < pi
16 X = original * (-1)**flip
17
18 T = 500000
19 J = 1
20 beta = 0.9
21
22 Y = X.copy()
23
24 for t in range(T):
25     i,j = np.random.choice(m), np.random.choice(n)
26
27     neighbors = [Y[i+a[0], j+a[1]] for a in
28                 [(-1,0), (1,0), (0,-1), (0,1)]
29                 if (0 <= i+a[0] < m) and (0 <= j+a[1] < n)]
30
31     alpha = min(np.exp(-2*beta*J*Y[i,j]*np.sum(neighbors)),1)
32
33     if np.random.rand() < alpha:
34         Y[i,j] = - Y[i,j]
35
36 fig, ax = plt.subplots(1,3, figsize=(24,16))
37 ax[0].imshow(original, cmap='gray', vmin=-1, vmax=1)
38 ax[1].imshow(X, cmap='gray', vmin=-1, vmax=1)
39 ax[2].imshow(Y, cmap='gray', vmin=-1, vmax=1)
40 plt.show()
```

# Literatura

- [1] S. BROOKS, A. GELMAN, G. L. JONES, X. MENG, *Handbook of Markov Chain Monte Carlo*, Chapman & Hall/CRC, Boston, 2011.
- [2] M. HOLMES–CERFON, *Applied Stochastic Analysis (web materijali)*, New York University, Warren Weaver Hall, 2019.  
Dostupno na:  
<https://tinyurl.com/eutbjr32>
- [3] S. I. RESNICK, *Adventures in Stochastic Processes*, Birckhauser, 2005.
- [4] N. SARAPA, *Teorija vjerojatnosti*, Školska knjiga, Zagreb, 1986.
- [5] Z. VONDRAČEK, *Markovljevi lanci (web materijali)*, PMF - Matematički odsjek, Sveučilište u Zagrebu, 2013.  
Dostupno na:  
<https://tinyurl.com/38f5ya8w>
- [6] A Conceptual Introduction to Markov Chain Monte Carlo Methods  
Dostupno na:  
<https://arxiv.org/pdf/1909.12313.pdf>
- [7] Advanced Statistical Computing, Metropolis-Hastings  
Dostupno na:  
<https://bookdown.org/rdpeng/advstatcomp/metropolis-hastings.html>
- [8] A survey of Monte Carlo methods for parameter estimation.  
Dostupno na:  
<https://d-nb.info/1214997430/34>
- [9] Examples of Adaptive MCMC  
Dostupno na:  
<http://probability.ca/jeff/ftplib/adaptex.pdf>
- [10] fiveMinuteStats, Introduction to Gibbs Sampling  
Dostupno na:  
<https://stephens999.github.io/fiveMinuteStats/gibbs1.html>
- [11] Investopedia, Monte Carlo Simulation: History, How it Works, and 4 Key Steps  
Dostupno na:  
<https://www.investopedia.com/terms/m/montecarlosimulation.asp>

- 
- [12] Minitab, The 4 Simple Steps for Creating a Monte Carlo Simulation with Engage or Workspace  
Dostupno na:  
<https://tinyurl.com/mryk99mf>
- [13] Monte Carlo Simulation to calculate the area under the curve.  
Dostupno na:  
<https://tinyurl.com/4ww2b2tc>
- [14] StatLect, Metropolis–Hastings algorithm  
Dostupno na:  
<https://tinyurl.com/2p8tahfe>
- [15] ResearchGate, Thermalisation and Relaxation of Quantum Systems  
Dostupno na:  
<https://tinyurl.com/2s4z5ywv>
- [16] Wikipedia, Ising model  
Dostupno na:  
[https://en.wikipedia.org/wiki/Ising\\_model](https://en.wikipedia.org/wiki/Ising_model)
- [17] Wikipedia, Monte Carlo method  
Dostupno na:  
[https://en.wikipedia.org/wiki/Monte\\_Carlo\\_method](https://en.wikipedia.org/wiki/Monte_Carlo_method)



# Sažetak

U ovom radu obradili smo primjenu Monte Carlo metode na Markovljevim lancima. Monte Carlo Markovljevi lanci (MCMC) metoda postala je temelj mnogih modernih znanstvenih analiza. Riječ je o stohastičkoj proceduri koja više puta generira slučajne uzorke koji karakteriziraju distribuciju od interesa. Proces generiranja slučajnih uzoraka u MCMC-u uloga je Markovljevog lanca, dok je proces generiranja statistike iz tih slučajnih uzoraka uloga Monte Carlo metode. Osvrnuli smo se na algoritme koji se najčešće koriste u praksi, a radi se o Metropolis–Hastings algoritmu te Gibbsovom uzorkovanju. Na kraju rada smo promotrili primjenu MCMC metode u domeni obrade slika, a riječ je o redukciji šuma sa slike.

## Ključne riječi

Monte Carlo metoda, Markovljevi lanci, Metropolis–Hastings algoritam, Gibbsovo uzorkovanje, Učinkovita veličina uzorka



# Markov chain Monte Carlo

## Summary

In this paper, we discussed the application of the Monte Carlo method on Markov chains. The Monte Carlo Markov Chain (MCMC) method has become the basis of many modern scientific analyses. It's a stochastic procedure that repeatedly generates random samples that characterize the distribution of interest. The process of generating random samples in MCMC is the role of a Markov chain, while the process of generating statistics from these random samples is the role of the Monte Carlo method. We looked at the algorithms that are most often used in practice, namely the Metropolis-Hastings algorithm and Gibbs sampling. At the end of the paper, we looked at the application of the MCMC method in the domain of image processing, which is the reduction of image noise.

## Keywords

Monte Carlo method, Markov chain, Metropolis-Hastings algorithm, Gibbs sampling, Effective Sample Size



# Životopis

Rođena sam 24. studenog 1998. godine u Vinkovcima. Pohađala sam Osnovnu školu "Mato Lovrak" u Županji te sam nakon nje upisala Opću gimnaziju u Županji. Srednju školu završila sam 2017. godine te sam iste godine upisala preddiplomski studij matematike i računarstva na Odjelu za matematiku u Osijeku. Preddiplomski studij završila sam 2020. godine sa završnim radom na temu "Slučajni vektor" pod mentorstvom izv. prof. dr. sc. Dragane Jankov – Maširević. Iste godine upisala sam diplomski studij na Odjelu za matematiku, smjer matematika i računarstvo. Stručnu praksu odradila sam u IT kompaniji Be – terna u Osijeku, gdje sam trenutno zaposlena na poziciji aplikacijskog inženjera.