Računarska statistika

Snježana Lubura Strunjak

Zagreb, 18. ožujka 2021.

Monte Carlo tehnike

Monte Carlo

- Monte Carlo - odnosi se na bilo koju proceduru koja koristi slučajne brojeve.

Monte Carlo

- Monte Carlo - odnosi se na bilo koju proceduru koja koristi slučajne brojeve.
- Niz slučajnih brojeva je skup brojeva koji nisu ni u kakvoj vezi sa ostalim brojevima u tom nizu.

Monte Carlo

- Monte Carlo - odnosi se na bilo koju proceduru koja koristi slučajne brojeve.
- Niz slučajnih brojeva je skup brojeva koji nisu ni u kakvoj vezi sa ostalim brojevima u tom nizu.
- Pseudo-slučajni brojevi su slučajni brojevi koje generiramo pomoću kompjuterskih algoritama, čine se slučajnim, ali su zapravo deterministički.

Monte Carlo

- Monte Carlo - odnosi se na bilo koju proceduru koja koristi slučajne brojeve.
- Niz slučajnih brojeva je skup brojeva koji nisu ni u kakvoj vezi sa ostalim brojevima u tom nizu.
- Pseudo-slučajni brojevi su slučajni brojevi koje generiramo pomoću kompjuterskih algoritama, čine se slučajnim, ali su zapravo deterministički.
- Niz pseudoslučajnih brojeva generiran kompjuterskim algoritmom je određen *seed*-om, što je početno stanje programa. Isti seed će svaki put generirati isti niz pseudo-slučajnih brojeva.

Generiranje uniformnih slučajnih brojeva

Uniformna distribucija - Svaki broj iz intervala [0,1] ima jednaku šansu (vjerojatnost) pojavljivanja.

Generiranje uniformnih slučajnih brojeva

Uniformna distribucija - Svaki broj iz intervala [0,1] ima jednaku šansu (vjerojatnost) pojavljivanja.

Primjer 1 (Algoritam srednjih kvadrata - J. Von Neumann, 1946.) Generiranje niza 10-znamenkastih brojeva:

1. počni s nekim 10-znamenkastim brojem
2. kvadriraj ga
3. uzmi srednjih 10 znamenaka kao sljedeći broj u nizu

Npr. $3690295441^2 = 13618280441865344481$, i za idući broj uzmemo 2804418653.
Problemi:

- Niz nije slučajan (svaki broj je u potpunosti određen prethodnim)
- Čini se slučajan, ali može ući u petlju, npr. za 4-znamenkasti broj:

$$6100^2 = 37210000 \to 2100^2 = 4410000 \to 4100^2 = 16810000 \to 8100^2 = 65610000$$
$$\to 6100^2 = \dots$$

Primjeri - generiranje pseudoslučajnih brojeva

Primjer 2 (Linearna kongruentna metoda - Lehmer, 1948.) Ako je $a \mod b$ ostatak pri djeljenju broja a s brojem b, onda je metoda dana formulom

$$I_{n+1} = (a \cdot I_n + c) \mod m,$$

gdje su:

- I_0 - početna vrijednost (sjeme(seed))
- $a, c \geq 0$
- $m > I_0, a, c$

Primjeri - generiranje pseudoslučajnih brojeva

Primjer 2 (Linearna kongruentna metoda - Lehmer, 1948.) Ako je $a \mod b$ ostatak pri djeljenju broja a s brojem b, onda je metoda dana formulom

$$I_{n+1} = (a \cdot I_n + c) \mod m,$$

gdje su:

- I_0 - početna vrijednost (sjeme(seed))
- $a, c \geq 0$
- $m > I_0, a, c$

Problem: Loš izbor konstanti može dovesti do loših nizova (možemo ući u petlju), npr. $a = c = I_0 = 7, m = 10 \rightarrow 7, 6, 9, 0, 7, 6, \ldots$

Primjeri - generiranje pseudoslučajnih brojeva

Primjer 2 (Linearna kongruentna metoda - Lehmer, 1948.) Ako je $a \mod b$ ostatak pri djeljenju broja a s brojem b, onda je metoda dana formulom

$$I_{n+1} = (a \cdot I_n + c) \mod m,$$

gdje su:

- I_0 - početna vrijednost (sjeme(seed))
- $a, c \geq 0$
- $m > I_0, a, c$

Problem: Loš izbor konstanti može dovesti do loših nizova (možemo ući u petlju), npr. $a = c = I_0 = 7, m = 10 \to 7, 6, 9, 0, 7, 6, \ldots$
Ako je $c = 0$, onda imamo multiplikativan kongruentan generator.

Primjeri - generiranje pseudoslučajnih brojeva

Primjer 2 (Linearna kongruentna metoda - Lehmer, 1948.) Ako je $a \mod b$ ostatak pri djeljenju broja a s brojem b, onda je metoda dana formulom

$$I_{n+1} = (a \cdot I_n + c) \mod m,$$

gdje su:

- I_0 - početna vrijednost (sjeme(seed))
- $a, c \geq 0$
- $m > I_0, a, c$

Problem: Loš izbor konstanti može dovesti do loših nizova (možemo ući u petlju), npr. $a = c = I_0 = 7, m = 10 \to 7, 6, 9, 0, 7, 6, \dots$
Ako je $c = 0$, onda imamo multiplikativan kongruentan generator.
Pravila za odabir konstanti a, c i m su dobro razvijena.

RANDU generator

U 1960-ima je IBM distribuirao popularan generator RANDU:

$$I_{n+1} = (65539 \cdot I_n) \mod 2^{31}.$$

Pomoću njega dobijemo pseudoslučajne cijele brojeve koji prate diskretnu uniformnu distribuciju na $\{1, 2, \ldots, 2^{31} - 1\}$.
Kasnije se pokazalo da ima ozbiljan problem.

RANDU generator

U 1960-ima je IBM distribuirao popularan generator RANDU:

$$I_{n+1} = (65539 \cdot I_n) \mod 2^{31}.$$

Pomoću njega dobijemo pseudoslučajne cijele brojeve koji prate diskretnu uniformnu distribuciju na $\{1, 2, \ldots, 2^{31} - 1\}$.
Kasnije se pokazalo da ima ozbiljan problem.
Program *CHAPTER*1_1_*RANDU*.*sas*. (n=20000, seed=45813)

RANDU generator

U 1960-ima je IBM distribuirao popularan generator RANDU:

$$I_{n+1} = (65539 \cdot I_n) \mod 2^{31}.$$

Pomoću njega dobijemo pseudoslučajne cijele brojeve koji prate diskretnu uniformnu distribuciju na $\{1, 2, \ldots, 2^{31} - 1\}$.
Kasnije se pokazalo da ima ozbiljan problem.
Program *CHAPTER1_1_RANDU.sas*. (n=20000, seed=45813)

Zadatak. (RANDU s različitim seed-ovima) Koristite program *CHAPTER1_1_RANDU.SAS*
Promjenite vrijednost makro varijable SEED na proizvoljno odabranu vrijednost.
Ispitajte grafički (eksploracijom) nezavisnost generiranih podataka.

Uvjeti za "dobar" uniformni generator slučajnih brojeva:

- uniformna marginalna distribucija
- nezavisnost
- ponovljivost(reproducibilnost) i prenosivost(na razne op. sisteme)
- brzina računanja

Ima mnogo statističkih i empirijskih testova za testiranje prva dva uvjeta. Dva skupa testova za generatore sl. brojeva:

1. DIEHARD ($http : //www.staff.science.uu.nl/ \sim sleij101/Opgaven/LabClass/site/asm\_diehard.php$)
(MarsagliaRN CDROM i DIEHARD battery of tests of randomness, 1985, 1995)
2. NIST Test suite (2000)
($http : //csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800 - 22b.pdf$)

Kako poboljšati ponašanje generatora slučajnih brojeva?

- Modifikacije:
$$I_n = (a \cdot I_{n-1} + b \cdot I_{n-2}) \mod m$$
koristi 2 početna seed-a i ima period $> m$.
- Treba koristiti generatore sa dokumentiranim svojstvima.
- U ovom kolegiju ćemo koristiti SAS i generatore slučajnih brojeva (kraće SB) u SAS-u

Generatori SB u SAS-u

- RANUNI (UNIFORM) funkcija
  - vraća broj generiran po uniformnoj distribucji na intervalu $[0, 1]$, koristi multiplikativni generator sa modulom $m = 2^{31} - 1$ i multiplikatorom $a = 397204094$ (Fishman and Moore 1982.)
  - testiran je i pokazalo se da ima dobro ponašanje (dužinu ciklusa 2147483646)
- RAND funkcija
  - koristi Mersenne-Twister generator SB (Matsumoto and Nishimura (1998), ACM Transactions on Modeling and Computer Simulation, 8, 3-30.)
  - ima period od $2^{19937} - 1$.

SAS-ov generator SB (RANUNI) - ponovljiv niz

Slika: Program *CHAPTER1_1_UNIFORM1.SAS*

```

/* CHAPTER1_1_UNIFORM1.SAS */

/** Generating a REPRODUCIBLE sequence of 100 (uniform) random numbers **/

DATA UNIFORM;
 DO REP = 1 TO 100;
  X = UNIFORM (1235);
  OUTPUT;
 END;
RUN;

/** or, more general (using macro variables) **/

%LET SEED =1235;
%LET NREP=1000;

DATA UNIFORM;
 DO REP = 1 TO &NREP;
  X = UNIFORM (&SEED);
  xlag=lag(x);
  OUTPUT;
 END;
RUN;

```

SAS-ov generator SB (RANUNI) - neponovljiv niz

Slika: Program *CHAPTER1_1_UNIFORM2.SAS*

```
/* CHAPTER1_1_UNIFORM2.SAS */

/** Generating an UNREPRODUCIBLE sequence of 100 (uniform) random n. **/

DATA RANDOM;
 DO REP = 1 TO 100;
  X = UNIFORM (0);
  OUTPUT;
 END;
RUN;

/** or, better **/

%LET SEED =0;
%LET NREP=100;

DATA RANDOM;
 DO REP = 1 TO &NREP;
  X = UNIFORM (&SEED);
  OUTPUT;
 END;
RUN;
```

SAS-ov generator SB (RAND) - ponovljiv niz

Slika: Program *CHAPTER1_1_UNIFORM3.SAS*

```
/* CHAPTER1_1_UNIFORM3.SAS */

/** Generating a REPRODUCIBLE sequence of 100 (uniform) random numbers **/

%LET SEED =1235;
%LET NREP=100;

DATA UNIFORM;
 CALL STREAMINIT(&SEED);
 DO REP = 1 TO &NREP;
  X = RAND ("UNIFORM");
 OUTPUT;
 END;
RUN;
```

Zadatak (Grafičko ispitivanje RANUNI (UNIFORM) generatora) Koristite program *CHAPTER1_1_RANDU.SAS*
Zamijenite RANDU generator sa SAS-ovim UNIFORM generatorom i ime dataseta promijenite u UNIFORM. Nakon toga ispitajte grafički (eksploracijom) nezavisnost generiranih podataka.

Generiranje ne-uniformnih brojeva

- Do sada jednostavne simulacije u kojima smo generirali nizove slučajnih brojeva (SB) uniformno distribuirane na intervalu [0,1].

Generiranje ne-uniformnih brojeva

- Do sada jednostavne simulacije u kojima smo generirali nizove slučajnih brojeva (SB) uniformno distribuirane na intervalu [0,1].
- Složeniji problemi zahtjevaju slučajne brojeve generirane po specifičnim distribucijama.

Generiranje ne-uniformnih brojeva

- Do sada jednostavne simulacije u kojima smo generirali nizove slučajnih brojeva (SB) uniformno distribuirane na intervalu [0,1].
- Složeniji problemi zahtjevaju slučajne brojeve generirane po specifičnim distribucijama.
- Npr., ako želimo proučavati osjetljivost t statistike na odstupanje od normalnosti, moramo generirati slučajne brojeve po normalnoj, gamma, itd. distribuciji.

Generiranje ne-uniformnih brojeva

- Do sada jednostavne simulacije u kojima smo generirali nizove slučajnih brojeva (SB) uniformno distribuirane na intervalu [0,1].
- Složeniji problemi zahtjevaju slučajne brojeve generirane po specifičnim distribucijama.
- Npr., ako želimo proučavati osjetljivost t statistike na odstupanje od normalnosti, moramo generirati slučajne brojeve po normalnoj, gamma, itd. distribuciji.
- Slučajni brojevi distribuirani po posebnim razdiobama, kao što je normalna, mogu se generirati uz pomoć posebnih algoritama.

Generiranje ne-uniformnih brojeva

- Do sada jednostavne simulacije u kojima smo generirali nizove slučajnih brojeva (SB) uniformno distribuirane na intervalu [0,1].
- Složeniji problemi zahtjevaju slučajne brojeve generirane po specifičnim distribucijama.
- Npr., ako želimo proučavati osjetljivost t statistike na odstupanje od normalnosti, moramo generirati slučajne brojeve po normalnoj, gamma, itd. distribuciji.
- Slučajni brojevi distribuirani po posebnim razdiobama, kao što je normalna, mogu se generirati uz pomoć posebnih algoritama.
- U MC studijama je često potrebno generirati nenormalne varijable sa zadanim vrijednostima koeficijenata asimetrije i spljoštenosti.

Metode za generiranje neuniformnih podataka

Metoda inverzne vjerojatnosti

Neka je X neprekidna slučajna varijabla s funkcijom distribucije F_X. Tada slučajna varijabla $U = F_X(X)$ ima $U(0,1)$ razdiobu.

Ovo nam daje vezu: $X = F_X^{-1}(U)$, gdje je U slučajna varijabla s $U(0,1)$ razdiobom, a F_X^{-1} je inverz funkcije distribucije slučajne varijable X kojeg moramo moći izračunati na intervalu $\langle 0, 1\rangle$.

Metoda inverzne vjerojatnosti za generiranje podataka po exponencijalnoj distribuciji

Neka je $X \sim Exp(\lambda)$, $\lambda > 0$. Tada je

$$F_X(x) = 1 - e^{-\lambda x},$$

za $x \geq 0$, a 0 inače.
Sada iz $y = F_X(x) = 1 - e^{-\lambda x}$ dobijemo $x = -\frac{1}{\lambda}\ln(1-y) = F_X^{-1}(y)$. Iz ovoga zaključujemo:

$$U \sim U(0,1) \Rightarrow X = F_X^{-1}(U) \sim Exp(\lambda).$$

Zadatak (Generiranje podataka po exponencijalnoj distribuciji metodom inverzne vjerojatnosti) Izvedite program

CHAPTER1_1_EXPO_INVERSION.SAS

U datasetu EXPONENTIAL ispitajte grafički i uz pomoć Kolmogorov-Smirnovog testa slaganje distribucije varijable *X_EXPO* sa eksponencijalnom distribucijom.
Uputa: Za grafički prikaz koristite Tasks and Utilities→ Statistics→ Distribution Analysis, i izaberite WORK.EXPONENTIAL i za varijablu izaberite *X_EXPO*. Za KS test koristite proceduru:

Slika: KS-test za eksponencijalnu razdiobu i grafički prikaz

```
proc univariate data=EXPONENTIAL;
    var X_EXPO;
    histogram X_EXPO/ exponential kernel;
run;
```

Može li se odbaciti nulta hipoteza da uzorak potječe iz populacije koja slijedi eksponencijalnu distribuciju?

SAS-ov eksponencijalni generator SB

Slika: Program *CHAPTER1_1_EXPO.SAS*

```
/** CHAPTER1_1_EXPO.sas **/

/** Generating a REPRODUCIBLE sequence of 100 random numbers from **/
/** EXPONENTIAL distribution **/

%LET SEED =1235;
%LET NREP=100;

DATA EXPONENTIAL;
 DO REP = 1 TO &NREP;
  X = RANEXP (&SEED);
  OUTPUT;
 END;
RUN;

/** or **/

DATA EXPONENTIAL;
CALL STREAMINIT(&SEED);
DO REP = 1 TO &NREP;
  X = RAND ("EXPO");
  OUTPUT;
 END;
RUN;
```

Metoda inverzne vjerojatnosti se može lako primijeniti za generiranje slučajnih brojeva po raznim diskretnim distribucijama.
Kontinuirane distribucije (osim eksponencijalne), kao npr. normalna i gamma distr. :

- nemaju jednostavnu funkcionalnu formu za inverz funkcije distribucije
- postoje aproksimacije inverza funkcija distribucije