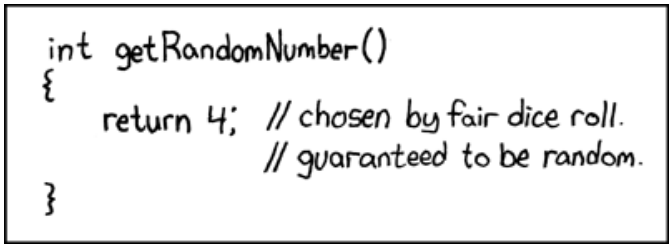
Generatorji slučajnih števil

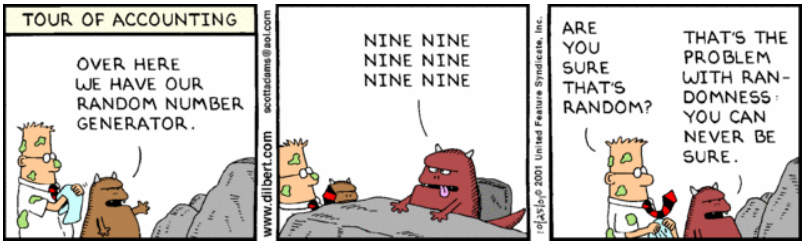
Miha Čančula

2. december 2011

Povzetek

The generation of random numbers is too important to be left to chance. – Robert R. Coveyou, *Oak Ridge National Laboratory*





1 Enakomerne porazdelitve

Najenostavnejša je bila primerjava med generatorji, ki vrnejo naključna števila, ki so enakomerno razporejena po nekem intervalu. Takšnih generatorjev je tudi največ, zato sem jih lahko primerjal. Za vsakega sem izračunal porazdelitev verjetnosti z razdelitvijo v B predalčkov, nato pa na ti porazdelitvi izvajal statistične teste. Uporabil sem naslednje generatorje:

1. Funkcija `rand()` iz standardne knjižnice jezika `C`
2. Linuxova datoteka `/dev/urandom`
3. Kalkulatorski generator, opisan v navodilih
4. "Mersennov vrtinec", implementiran v `GSL`

Poleg teh generatorjev sem za primerjavo vključil še dva izvora naključnih števil, ki števil ne generirata s številskim algoritmom, ampak jih izračunata težko predvidljivih dogodkov. V nasprotju z generatorji na ta način ne moremo dobiti poljubnega števila naključnih števil, zato sem velikost vzorca omejih na 4096 števil, kjer ima vsako število 32 bitov.

1. Linuxova datoteka `/dev/random`, ki naključnost dobi iz dogodkov v računalniku, na primer s premikanjem miške

2. Spletna stran `random.org`, ki naključnost dobi iz meritev atmosferskega šuma

Žal se je izkazalo, da je branje iz `/dev/random` prepočasno za kakršnekoli statistične teste, saj sem le po divjem mahanju z miško uspel dobiti 100 naključnih števil. To je dovolj za generiranje naključnih gesel in šifrirnih ključev, ne pa za statistiko, zato sem ta generator izpustil iz nadaljnjih računov.

Za vse statistične teste χ^2 sem v tabelo vpisal verjetnosti, poleg tega pa sem dodal še pričakovane vrednosti za idealen generator naključnih števil. Testu Kolmogorov-Smirnova sem opravil brez predalčkanja, tako da sem generirana števila najprej uredil, nato pa njihovo kumulativno porazdelitev primerjal s pričakovano. Vpisal sem dobljeno vrednosti statistike D, ki bi pri stopnji tveganja 5% morala biti nižja od 1,36. Pri enodimenzionalnih testih sem podatke razdelil v $\sqrt{N}$ predalčkov, pri dvodimenzionalnih pa v $\sqrt[3]{N} \times \sqrt[3]{N}$ predalčkov. Z rdečo barve sem označil vrednosti, ki bi jih lahko zavrgli z manj kot 5% tveganjem.

Test	1D χ^2			2D χ^2			$D\sqrt{N}$			Čas [s]
N	2^{12}	2^{18}	2^{24}	2^{12}	2^{18}	2^{24}	2^{12}	2^{18}	2^{24}	2^{28}
`rand()`	35%	32%	34%	65%	57%	29%	0,52	1,22	0,59	3.3
`/dev/urandom`	54%	61 %	29%	21%	53%	84%	0,74	1,17	0,47	110
Kalkulatorski	74%	98%	0%	70%	4%	0%	0,49	0,69	15,2	36
Mersenne	17%	71%	63%	0,5%	50%	62%	0,90	1,07	0,63	2.9
`random.org`	28%			49%			0,90			
Idealno	50%			50%			1			

Tabela 1: Statistični testi enakomernosti naključnih števil

Pri obeh testih χ^2 si želimo verjetnost čim bližje 50%. Premajhna vrednosti (npr. manj kot 5%) pomeni, da lahko generator zavržemo z majhno stopnjo tveganja, saj števila niso porazdeljena dovolj enakomerno. Podobno pa prevelika vrednost (npr. več kot 95%) pomeni, da je porazdelitev preveč enakomerna, saj so odstopanja manjša od pričakovane statistične napake. Oboje namreč pomeni, da števila niso naključna.

Iz podatkov lahko zaključimo, da je izmed preizkušanih generatorjev najboljši Mersennov vrtinec. Izkaže se predvsem, če moramo generirati veliko naključnih števil, tako z dobrimi rezultati testov kot tudi s hitrostjo. Pri manjših vzorcih je verjetno enostavneje izbrati vgrajeni generator, ki se ponaša s podobno hitrostjo, in celo z boljšim χ^2 za dovolj majhne N. Kalkulatorski generator je v vseh primerih najslabši.

2 Smeri v prostoru

Za generacijo naključnih meri v prostoru najprej potrebujemo generator enakomernih naključnih števil. V ta namen sem uporabil kar najboljši generator iz prve naloge, to je bil Mersennov vrtinec. Knjižnica `GSL` ima vgrajeno rutino, ki s pomočjo enakomernega generatorja vrne naključen enotski vektor v treh dimenzijah. Kljub temu pa sem za primerjavo še sam napisal takšen generator. Za primer brez sevanja je algoritem enostaven, saj vemo da mora biti porazdelitev verjetnosti enakomerna po spremenljivkah φ in $\cos\vartheta$. Če sta r_1 in r_2 dve naključni števili med 0 in 1, lahko prostorski kot zapišemo kot

$$\varphi = 2\pi r_1 \tag{1}$$
$$\vartheta = \arccos(2r_2 - 1) \tag{2}$$

Paziti moramo, da je $r_1 \in [0, 1)$ in $r_2 \in [0, 1]$, saj kota $\varphi = 0$ in $\varphi = 2\pi$ predstavljata isto koordinato na krogli, medtem ko sta $\vartheta = 0$ in $\vartheta = \pi$ različni točki. Zaradi omejene natančnosti strojnega računanja sem to moral upoštevati pri generiranju števil.

Pri dipolnem sevanju je porazdelitev verjetnosti po kotu ϑ bolj zapletena.

$$\frac{\partial p}{\partial \vartheta} \propto \sin^3\vartheta \tag{3}$$
$$\mathrm{d}p = C\sin^3\vartheta\,\mathrm{d}\vartheta = C(1-\cos^2\vartheta)\,\mathrm{d}(\cos\vartheta) \tag{4}$$
$$= C\,\mathrm{d}\left(\cos\vartheta - \frac{\cos\vartheta^3}{3}\right) \tag{5}$$

Če upostevamo, da je r_2 enakomerno razporejen, dobimo enačbo za ϑ.

$$\frac{\cos^3\vartheta}{3} - \cos\vartheta + Ar_2 + B = 0 \tag{6}$$

Da izrazimo $\cos\vartheta$ s pomočjo r_2 bomo morali rešiti polinom tretje stopnje. Enačba mora vedno imeti eno rešitev za $\cos\vartheta$ na intervalu $[-1, 1]$. Konstanti A in B torej določimo tako, da bo preslikava med $\cos\vartheta \in [-1, 1]$ in $r_2 \in [0, 1]$ bijektivna. To došezemo, če je

$$\frac{\cos^3\vartheta}{3} - \cos\vartheta + \frac{2}{3}(2r_2 - 1) = 0 \tag{7}$$

V tem primeru ima polinom vedno tri realne korene, od katerih je srednji med -1 in 1.

2.1 Krogelne funkcije

Vrednosti za ϑ in φ nisem razdelil v histogram, ampak sem izračunal pričakovane vrednosti nekaterih osnovnih momentov. Računal sem povprecji φ in $\cos\theta$, povprečen kvadrat $\cos^2\theta$, poleg tega pa se krogelno funkcijo $Y_1^1 = \cos\varphi\cos\theta$, ki preverja korelacijo med obema spremenljivkama. Vse račune sem izvajal z $N = 2^{24}$ števili. Rezultati so v tabeli 2.

Generator	$\langle \cos\vartheta \rangle$	$\langle \varphi \rangle$	$\langle \cos^2\vartheta \rangle$	$\langle Y_1^1 \rangle$
Naključna smer (`GSL`)	$-3,88 \cdot 10^{-5}$	$14,0 \cdot 10^{-5}$	0.333326	$5.58 \cdot 10^{-7}$
Naključna smer (jaz)	$5,96 \cdot 10^{-5}$	$-25,9 \cdot 10^{-5}$	0.333356	$4,86 \cdot 10^{-5}$
Pričakovano	0	0	1/3	0

Tabela 2: Primerjava osnovnih momentov za obe implementaciji naključne smeri v prostoru.

Vidimo, da je moja implementacija generatorja naključnih smeri, ki sledi enačbama (1) in (2), slabša od algoritma v `GSL`. Odstopanje od pričakovanega povprečja je bilo večje v vseh treh kriterijih, kjer upoštevamo le eno spremenljivko. Po drugi strani pa je korelacija med φ in ϑ manjša, kar lahko razložimo s tem, da v zgornjih enačbah ni povezave med ϑ in φ.

Generator	$\langle \cos\vartheta \rangle$	$\langle \varphi \rangle$	$\langle \cos^2\vartheta \rangle$	$\langle Y_1^1 \rangle$
Dipolno sevanje (jaz)	$-2,28 \cdot 10^{-6}$	$12,9 \cdot 10^{-5}$	0,199994	$-1,20 \cdot 10^{-5}$
Pričakovano	0	0	1/5	0

Tabela 3: Nekaj osnovnih momentov za dipolno sevanje

Odstopanje povprečne vrednosti $\cos^2\theta$ je podobnega velikostnega reda kot ostala odstopanja od povprečij, tako da lahko zaključim, da generator deluje pravilno.

3 Gaussova porazdelitev

Tu je postopek podoben kot pri generiranju naključnih smeri. Uporabimo generator enakomerno porazdeljenih števil, ki jih transformiramo na tak način, da bo porazdelitev transformirank Gaussova. V ta namen se največ uporabljata Box-Mullerjeva transformacija.

Obstajajo pa tudi drugi pristopi, od katerih je najbolj uporabljana metoda "zigurat", ki je tudi najhitrejša. Uporabil sem obe, poleg teh pa sem dodal še metodo "ratio-of-uniforms" Kindermanna in Monahana, ki je prav tako implementirana v knjižnici `GSL`.

Test	1D χ^2			2D χ^2			$D\sqrt{N}$			Čas [s]
N	2^{12}	2^{18}	2^{24}	2^{12}	2^{18}	2^{24}	2^{12}	2^{18}	2^{24}	2^{24}
Box-Muller	78%	37%	84%	49%	25%	57%	0,85	1,06	1,58	43
Zigurat	81%	60%	32%	99%	60%	95%	0,62	0,83	1,08	6,5
Razmerje	80%	10%	23%	86%	62%	37%	0,51	1,07	0,86	16
Idealno	50%			50%			1			

Tabela 4: Statistični testi gaussovskih naključnih števil

Vse metode so počasnejše od generiranja enakomerno porazdeljenih števil. Najhitrejša je metoda "zigurat", ki pa se na statističnih testih ne izkaže dobro, saj bi jo na dvodimenzionalnem testu χ^2 lahko zavrgli brez prevelikega tveganja. Dober kompromis ponuja tudi metoda z razmerji, saj prestane vse teste, je pa še vedno hitrejša od Box-Mullerjeve metode.