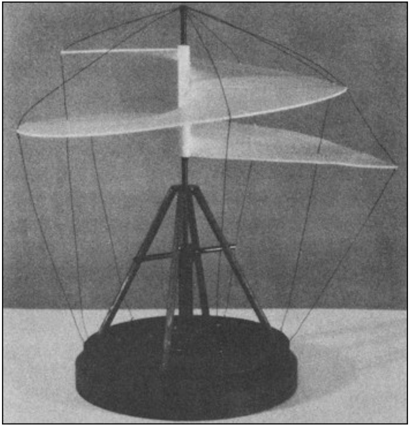
Statistične simulacije

Uvod

- Nekateri problemi so preveč kompleksni, da bi jih obravnavali analitično. Uporabimo lahko simulacije
  - Pretok prometa v križišču.
  - Simulacija pojavov, ki jih v naravi ni mogoče (dobro) spremljati: udar strele, vulkanski izburuh, tornado, poplava,...
  - Poenostavljene simulacije ekonomsko dragih projektov (vesoljske raziskave, raziskave v kemiji,...).
  - Simulacije iger z velikim številom dejavnikov vpliva.

- Simulacije večinoma temeljijo na uporabi računalnika. Poznamo pa tudi druge primere: Leonardo da Vincija in simulacija helikopterja, simulacije v bioloških raziskavah,...



Slika: Model helikopterja po da Vinciju (1493).

- Simulacija je umetno generiranje podatkov, ki mu sledi oblikovanje modela in nastavljanje parametrov.
- Kritična presoja rezultatov simulacije potrdi (vsaj deloma) model, ali ga ovrže in narekuje njegove spremembe.
- Večinoma je pomembno, da so podatki generirani naključno.
- Zato imajo odločilni pomeni generatorji naključnih števil.

Primer

Verjetnost meta šestice na klasični igralni kocki zlahka ocenimo statistično. Generiramo N naključnih metov kocke in preštejemo število metov šestice (n_6). Statistična verjtnost p_6 meta šestice je torej približno

$$p_6 \approx \frac{n_6}{N}.$$

Oceno še izboljšamo, če poskus ponovimo večkrat in za rezultat vzamemo povprečje.

- Očitno pomembno vlogo igra naključno generiranje podatkov.

Generatorji naključnih števil

Primer

Ista verzija programa Matlab (od verzije 7.4 naprej), bo na vseh računalnikih kot rezultat ukaza `rand(1)` takoj po zagonu vrnila `0.8147`. To naj bi bilo naključno število med 0 in 1???

- Računalniki so deterministični, zato lahko generirajo le psevdonaključna števila.
- To pomeni, da so "naključna" števila določena z nekimi začetnimi parametri.
- Njapogosteje se uporabljajo multiplikativni kongruenčni algoritmi.

- Lehmerjev algoritem: podani so celoštevilski parametri a, c in m ter začetna vrednost x_0. Tvorimo zaporedje števil

$$x_{k+1} = a\, x_k + c \mod m, \quad k = 0, 1, \ldots$$

- Števila lahko potem z deljenjem z m še normiramo na $[0, 1)$.
- Pri primerno izbranih parametrih se zdi zaporednje "naključno".
- Sestavljeno je seveda iz končnega števila možnih vrednost, kvečjemu m.

Primer

Če za parametre izberemo $a = 13$, $c = 0$, $m = 31$ *in* $x_0 = 1$, *dobimo zaporedje,*

$$1,\ 13,\ 14,\ 27,\ 10,\ 6,\ 16,\ 22,\ \ldots,$$

oziroma normirano na $[0, 1)$

$$0.0323,\ 0.4194,\ 0.4516,\ 0.8710,\ 0.3226,\ 0.1935.\ 0.5161,\ 0.7097,\ ,\ldots$$

Lahko se prepričamo, da ima zapordje periodo 30.

Primer

Med prvimi primeri uprabe tega generatroja je bil IBM v šestdesetih letih prejšnjega stoletja (t.i. Scientific Subroutine Package (SPS)). Izbrali so parametre $a = 65539$, $c = 0$ in $m = 2^{31}$.Ostanek se v 32 bitni dolžini preprosto računa, obenem pa je $a = 2^{16} + 3$, kar olajša računanje množenja. V takratni dobi je bil to zelo pomembno zaradi hitrosti računanja. Toda izbira ima tudi nezaželjene posledice. Ker je (vse po modulu 2^{31})

$$
\begin{aligned}
x_{k+2} &= \left(2^{16} + 3\right) x_{k+1} = \left(2^{16} + 3\right)^2 x_k \\
&= \left(2^{32} + 6 \cdot 2^{16} + 9\right) x_k \\
&= \left[6 \left(\cdot 2^{16} + 3\right) - 9\right] x_k .
\end{aligned}
$$

Torej je $x_{k+2} = 6\, x_{k+1} - 9\, x_k$, kar pomeni, da so tri zaporedna "naključna" števila povezana.

- Matlab je dolgo uporabljal Lehmerjev algoritem s parametri $a = 7$, $c = 0$ in $m = 2^{31} - 1$.
- Generira normirana števila med 0.00000000046566 in 0.99999999953434 s periodo $m - 1$.
- Kljub temu, da dobimo več kot 10^9 psevdonaključnih števil, je to za danešnje hitrosti računalnikov premalo.
- Matlab je zato od verzije 5.0 do 7.3 uporabljal algoritem, ki temelji na izsledkih Georgea Marsgalie.
- Od verzije 7.4 naprej pa uporablja Mersenne Twisterjev algoritem. Za podrobnosti si oglejte `help rand` ali stran http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/emt.html

- Očitno je pri izbiri parametrov a, c in m vse določeno z x_0.
- Številu x_0 rečemo seme.
- Matlab dovoljuje določitev semena pri posameznih generatorjih. Izberemo ga denimo iz računalniškega časa (clock).
- Obstajajo pa tudi generatorji, ki temeljijo na popolnoma drugačnih osnovah, recimo na fizikalnih pojavih: žarki sevanja, digitaliziran hrup, ...
- Vsi generatorji naključnih števil se ponavadi testirajo s standardnimi statističnimi testi, ki "ocenijo naključnost" generatorja.

Računanje števila π

Primer

Naključno izbiramo pare števil (x, y) *v kvadratu* $[0, 1] \times [0, 1]$. *Označimo z N število vseh izbir, s K pa število tistih izbir, za katere je* $x^2 + y^2 \leq 1$. *Pri predpostavki, da so števila "naključna", očitno velja*

$$\frac{K}{N} \approx \frac{\pi/4}{1},$$

Torej je

$$\pi \approx \frac{4\,K}{N}.$$

Primer

Prejšnji primer lahko posplošimo v d dimenzij. Izbirajmo sedaj d-terice naključnih števil med -1 *in* 1*, torej* $(x_1, x_2, \dots, x_d) \in [-1, 1]^d$*. Spet naj bo* N *število vseh izbir,* K *pa število tistih, za katere je* $x_1^2 + x_2^2 + \cdots + x_d^2 \leq 1$ *(točka* $(x_1, x_2, \dots, x_d)$ *leži v enotski d-dimenzionalni krogli v* $\mathbb{R}^d$*). Volumen d-dimenzionalne krogle je*

$$V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2} + 1\right)},$$

kjer je Γ *Eulerjeva gama funkcija. Za sode dimenzije* $d = 2\,k$ *se formula poenostavi v* $V_d = \pi^k / k!$*. Ker je volumen kocke* $[-1, 1]^d$ *enak* 2^d*, je ocena za* π *torej*

$$\pi \approx \left(2^d\,\Gamma\left(\frac{d}{2} + 1\right)\frac{K}{N}\right)^{\frac{2}{d}}.$$

Primer

Oglejmo si še nekoliko bolj zapleteno simulacijo, Buffonovo iglo. Problem je bil prvič zastavljen leta 1777.
Na "neskončnem" papirju so zarisane vzporedne vodoravne črte na razdalji 1. *Iglo dolžine* 1 *vržemo na papir. Kolikšna je verjetnost* p, *da bo igla presekala kakšno od črt?*
Če verjetnost p *izračunamo in naredimo simulacijo z* N *meti, od katerih jih* K *preseka kakšno od črt, potem velja*

$$p \approx \frac{K}{N}.$$

Izkaže se, da je $p = 2/\pi$, *za to je*

$$\pi \approx 2\frac{N}{K}.$$

Kaj pa dva preostala primera, ko je dolžina igle manjša kot razdalja med črtami, ali ko je dolžina igle večja od razdalje med črtami?

Računanje določenega integrala

- Numerično želimo izračunati integral

$$\int_a^b f(x)dx.$$

- Uporabimo lahko kakšnega izmed množice kvadraturnih pravil:
  - trapezno pravilo,
  - Simpsonovo pravilo,
  - Gauss-Legendreovo pravilo,
  - ...
- Lahko pa integral približno določimo s simulacijo.

Metoda "zadeni ali zgreši"

- Predpostavimo, da je $0 \leq f(x) \leq M$, $x \in [a,b]$. (Kaj če ni?)
- Izberemo N naključnih točk $(x,y) \in [a,b] \times [0,M]$.
- Denimo, da je K točk takih, za katere je $y \leq f(x)$.
- Potem je

$$\frac{K}{N} \approx \frac{\int_a^b f(x)\,dx}{(b-a)M}.$$

Torej je

$$\int_a^b f(x)\,dx \approx (b-a)\,M\,\frac{K}{N}.$$

Metoda "povprečne vrednosti"

- Izkoristimo dejstvo o povprečni vrednosti

$$f(\xi) = \frac{1}{b-a}\int_a^b f(x)\,dx, \quad \xi \in [a,b].$$

- Če interval $[a,b]$ razdelimo na N enakih podintervalov širine $h := (b-a)/N$, je na primer

$$\int_a^b f(x)\,dx \approx h\sum_{i=1}^{N} f(x_i), \quad x_i = a + (2i-1)\,h/2,\ i = 1, 2, \ldots, N.$$

- Namesto, da bi točke izbirali enakomerno, jih izbiramo naključno na $[a, b]$.
- Tako za vsak nabor N naključnih točk x_i dobimo oceno

$$\int_a^b f(x)\,dx \approx h \sum_{i=1}^{N} f(x_i).$$

- Metodo zlahka posplošimo na večkratne integrale:

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_d}^{b_d} f(x_1, x_2, \ldots, x_d)\,dx_d\,dx_{d-1} \ldots dx_1.$$

- Pri klasični intergaciji izberemo N_i naključnih točk $\xi_{i,j}$, $j = 1, 2, \ldots, N_i$ na $[a_i, b_i]$, $i = 1, 2, \ldots, d$ in izračunamo približek za integral

$$\frac{(b_1 - a_1) \cdots (b_d - a_d)}{N_1 \cdots N_d} \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \cdots \sum_{i_d=1}^{N_d} f(\xi_{1,i_1}, \ldots, \xi_{d,i_d}).$$

- V smislu metode Monte Carlo pa bi večkratni integral lahko izračunali tudi z izborom N naključnih točk v domeni $[a_1, b_1] \times \cdots \times [a_d, b_d]$, namreč

$$\frac{(b_1 - a_1) \cdots (b_d - a_d)}{N} \sum_{i=1}^{N} f(x_{1,i}, x_{2,i}, \ldots, x_{d,i}).$$

- Pri klasični numerični integraciji moramo v vsaki koordinatni smeri izbrati vsaj nekaj točk. Njihovo število raste kot $N_1 N_2 \cdots N_d$.
- Pri metodi Monte Carlo pa N raste neodvisno od d.
- Pokazati se da, da v tem primeru napaka (verjetnostno) pada z $\sqrt{N}$ neodvisno od d.