

# Tutkimustiedonhallinnan peruskurssi

Hannu Toivonen, Marko Salmenkivi, Inkeri Verkamo

`hannu.toivonen, marko.salmenkivi, inkeri.verkamo@cs.helsinki.fi`

Helsingin yliopisto – Tietojenkäsittelytieteen laitos

# Satunnaislukujen generointi



# Lähteet

---

- Knuth, D., The Art of Computer Programming, Volume 2: Seminumerical Algorithms, luku 3
- Numerical Recipes in C, luku 7

# Johdanto

---

- edellä olemme tutustuneet menetelmiin, jotka perustuvat satunnaislukujen generointiin
  - satunnaistustestaus: muodostetaan (kahden) otoksen permutaatioita
  - bootstrap: generoidaan pseudo-otoksia valitsemalla satunnaisesti alkioita otoksesta takaisinpanolla
  - Monte Carlo -testi usein: generoidaan arvoja tunnetusta populaatiojakaumasta

# Johdanto (2)

---

- satunnaistus ja bootstrap edellyttävät satunnaislukujen generoimista tasaisesta jakaumasta
  - tasaisesti jakautuneiden satunnaislukujen generointiin ohjelmointikielissä kirjastofunktiot
- Monte Carlo -testauksen kohdalla populaatiojakauma voi luonnollisesti olla mikä tahansa jakauma
  - kuinka generoidaan satunnaislukuja muista kuin tasaisista jakaumista?
  - vaikka populaatiojakauma tunnettaisiinkin, ei välttämättä ole aivan suoraviivaista generoida siitä otoksia

# Diskreettejä jakaumia

---

- satunnaismuuttuja  $X$  noudattaa aina jotakin todennäköisyysjakaumaa
- jos muuttujan arvot ovat diskreettejä jakauma on diskreetti
- diskreetin todennäköisyysjakauman määrittää pistetodennäköisyysfunktio  $p(x)$ , jolle  $\sum_i p(i) = 1$
- keskeisiä diskreettejä jakaumia:
  - diskreetti tasainen jakauma
  - binomijakauma
  - Poisson-jakauma
  - geometrinen jakauma

# Keskeisiä jatkuvia jakaumia

- muuttujan arvot jatkuvia  $\rightarrow$  jakauma jatkuva
- jatkuvan jakauman määrittää tiheysfunktio  $f(X = x)$ , jolle  $\int_{-\infty}^{\infty} f(x)dx = 1$
- tiheysfunktion integraalifunktiota  $F(x) = P(X \leq x)$  nimitetään jakaumafunktioksi (kertymäfunktioksi)
- keskeisiä jatkuvia jakaumia
  - tasainen jakauma,  
 $X \sim Tas(a, b) \Leftrightarrow f(X = x) = \frac{1}{b-a}, a < x < b$
  - normaalijakauma
  - eksponenttijakauma
  - gammajakauma

# Satunnaislukujen generointi

---

- *“If the numbers are not random, they are at least higgledy-piggledy (‘sikin sokin’)”*  
— GEORGE MARSAGLIA (1984)
- tietokoneen generoimat satunnaisluvut ovat pseudo-satunnaislukuja
- deterministisesti määräytyneitä, ei “oikeasti satunnaisia”
- mikä on riittävän satunnaista yhdelle sovellukselle ei välttämättä ole sitä toiselle
- olemassa tilastollisia testejä, joilla testataan satunnaisuutta



# Linear congruential method

---

- $m$  modulus
- $a$  kertoja
- $c$  lisäys
- $X_0$  alkuarvo
- satunnaislukusekvenssi saadaan rekursioyhtälöstä

$$X_{n+1} = (aX_n + c) \mod m, \quad n \geq 0$$

- yleisin tasaisesti jakautuneiden satunnaislukujen generoimismenetelmä
- parametrien  $m, a, c$  valinta on erittäin keskeinen kysymys toimivuuden kannalta

# Em. menetelmän ominaisuuksia

- menetelmä on syklinen
- sykliä nimitetään periodiksi
- sama siemenluku  $\rightarrow$  periodi alkaa samasta kohdasta  $\rightarrow$  sama sarja “satunnaislukuja”
- käytännössä siemenluvun voi ottaa esim. systeemin kellonajasta
- testausvaiheessa on hyvä että siemenluvun voi asettaa käsin

# Kertymäfunktion käänteisfunktio

- todennäköisyysjakauma voidaan siis ilmaista sen kertymäfunktion  $F(x) = P(X \leq x)$  avulla
- $F(x)$  on aina monotoninen (kasvava) funktio
  - siis  $F(x_1) \leq F(x_2)$ , jos  $x_1 \leq x_2$
  - $F$  saa arvot välillä  $[0, 1]$
- kun  $F(x)$  on jatkuva ja aidosti kasvava, sillä on käänteisfunktio  $F^{-1}(y)$ ,  $0 < y < 1$
- $y = F(x)$  jos ja vain jos  $x = F^{-1}(y)$
- käänteisfunktioita voidaan käyttää satunnaislukujen generoimiseen kyseisestä jakaumasta

# Transformaatiomenetelmä

- olkoon  $f(x)$  sen jakauman tiheysfunktio, josta halutaan generoida satunnaislukuja, ja  $F(x)$  vastaava kertymäfunktio
- oletetaan että kertymäfunktilla on käänteisfunktio  $x = F^{-1}(y), 0 < y < 1$
- transformaatiomenetelmä:
  1. generoi satunnaisluku  $y \sim Tas(0, 1)$
  2.  $x = F^{-1}(y)$  on satunnaisluku joka noudattaa haluttua jakaumaa

# Esimerkki: eksponenttijakauma

- keskeisin jakauma tasaisen jakauman ja normaalijakauman jälkeen
- jos tulostimelle saapuu keskimäärin  $\mu$  tulostuspyyntöä aikayksikköä kohti ja saapumisajat ovat satunnaisia, niiden välit  $x_i$  noudattavat eksponenttijakaumaa,  $X_i \sim \text{Exp}(\mu)$
- $p(X = x) = \mu e^{-\mu x}$
- $E(X) = 1/\mu, \text{ var}(X) = 1/\mu^2$

- 
- eksponenttijakauman kertymäfunktio:

$$F(x) = 1 - e^{-x/\mu}, x > 0$$

- $F^{-1}(y) = -\mu \log(1 - y)$

$$\Rightarrow -\mu \log(1 - U) \sim \text{Exp}(\mu)$$

- $1 - U$  on tasaisesti jakautunut silloin kun  $U$  on

$$\Rightarrow x = -\mu \log U \text{ on eksponenttijakautunut odotusarvolla } \mu$$

# Hylkäämismenetelmä

---

- *rejection sampling*
- myös hyväksymis-hylkäämismenetelmä
- ei edellytä kertymäfunktion eikä sen käänteisfunktion tuntemista
- on tunnettava apufunktio  $g(x)$ , joka saa kaikkialla suuremman arvo kuin  $f(x)$  josta otoksia halutaan
- $g$ :n alan oltava äärellinen

# Algoritmi

- olkoon  $g(x) \geq f(x)$ , kaikilla  $x \in \mathbb{R}$
- olkoon  $A = \int_{-\infty}^{\infty} g(x)dx$  ja  $G(x)$  vastaava integraalifunktio (“kertymäfunktio”) jolla on käänteisfunktio joka pystytään laskemaan
  1. generoi luku  $y \sim Tas(0, A)$
  2. kandidaatti  $x$  saadaan transformaatiomenetelmellä
$$x = G^{-1}(y)$$
  3. generoi  $Z \sim Tas(0, g(x))$
  4. **if**  $z < f(x)$  tulosta  $x$  **else goto** 1



# Toinen muotoilu menetelmälle

- oletetaan että

- (i)  $h$  on sellaisen jakauman  $H$  tiheysfunktio, josta voidaan generoida arvoja
- (ii)  $f(x) \leq c h(x)$ , kaikille  $x$ , jollakin vakiolla  $c$

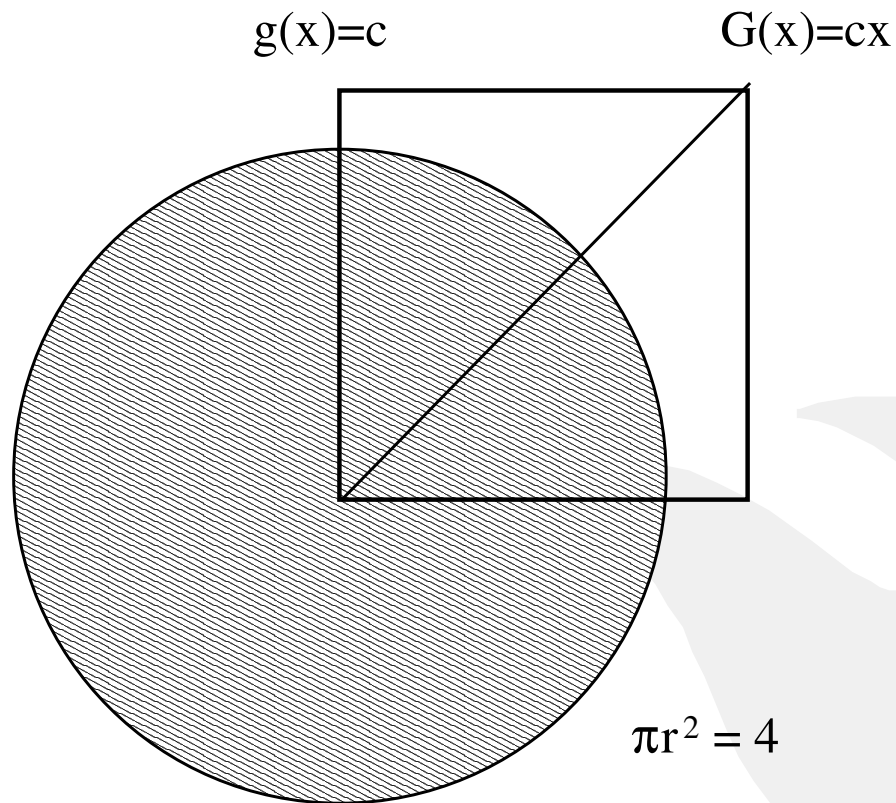
- Algoritmi

1. generoi kandidaatti  $z \sim H$
2. generoi  $u \sim \text{Unif}(0, 1)$
3. **if**  $u \leq f(z)/(ch(z))$  **return**  $z$ ; **else goto** 1

- tehokkuus riippuu suhteesta  $f(z)/ch(z)$  eli  $c$ :stä

# Hyvin simppleri esimerkki

- halutaan generoida satunnaislukuja jakaumasta, jonka tiheysfunktio on neljännesympyrän kehä



- $\pi r^2 / 4 = 1 \Rightarrow r = 2 / \sqrt{(\pi)}$

- 
- olkoon apufunktio  $g(x) = c, c > r \Rightarrow G(x) = cx$
  - 1. generoidaan satunnaisluku  $y \sim Tas(0, c^2)$
  - 2. määritetään kandidaatti  $x = G^{-1}(y) = y/c$
  - 3. generoidaan satunnaisluku  $u \sim Tas(0, g(x) = c)$
  - 4. jos  $f(x) = \sqrt{\left(\frac{4}{\pi} - x^2\right)} < u$  palautetaan  $x$  muuten palataan kohtaan 1

# Tosielämässä

---

- monista keskeisistä jakaumista voidaan generoida satunnaislukuja hylkäämismenetelmää käyttäen
- esim. normaali-, gamma-, Poisson-, binomijakaumat
- keskeinen apufunktio  $g$  on Lorentzin jakauma tiheysfunktio; kyseisen jakauman integraalifunktio on tangenttifunktio

# Arvojen riippumattomuus

---

- hylkäämismenetelmässä jokainen kandidaatti generoitiin siten, että peräkkäiset arvot eivät riippuneet toisistaan
- jos vakio  $c$  suuri, vain pieni osa “arvauksista” osuu oikeaan
- avaruus on liian “harva”, arvaukset osuvat liian harvoin
- hyvin monimutkaisista jakaumista (paljon ulottuvuuksia ja riippuvuuksia niiden välillä) ei riippumattomia arvoja voi generoida
- tilastollisessa mallintamisessa saatetaan tarvita otoksia hyvin “hankalista” jakaumista

# MCMC-menetelmä

---

- Markovin ketju Monte Carlo
- sallitaan, että peräkkäiset arvot ovat toisistaan riippuvia
- kun on tehty hyvä “arvaus”, kokeillaan seuraavaksi jotakin lähellä olevaa arvoa
- joka kerta kun generoidaan arvoa halutusta jakaumasta, otetaan huomioon edellinen arvo, mutta ei muita

# Markovin ketju

---

- *Markovin ketju* on jono satunnaismuuttujia siten, että jonon seuraava muuttuja riippuu edellisestä (mutta ei muista)

$$Pr(X_n | X_1, \dots, X_{n-1}) = Pr(X_n | X_{n-1})$$

- idea: generoidaan ketju, jonka tilojen jakauma on otos halutusta jakaumasta

# Algoritmi (Metropolis-Hastings)

- valitse alkutila  $x_0$
- toista  $K$  kertaa ( $n = 0, \dots, K - 1$ ):
  - generoi kandidaatti  $x'$  jostakin jakaumasta  $Q(x_n)$
  - todennäköisyydellä  $\alpha(x_n, x')$  aseta  $x_{n+1} := x'$
  - muussa tapauksessa aseta  $x_{n+1} := x_n$
  - aseta  $n := n + 1$
  - tulosta  $x_n$
- yllä  $Q(x)$  on  $x$ :n ehdotusjakauma
- $q(x, x')$  on  $Q$ :n tiheysfunktio

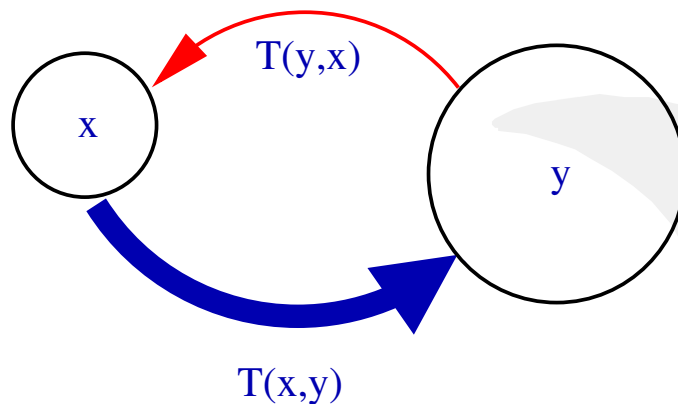


# Käänteisyysehto

- M-H -algoritmi perustuu *käänteisyysehtoon*
- jos kaikille pareille  $x, y \in E$  pätee

$$f(x) T(x, y) = f(y) T(y, x)$$

$\Rightarrow$  ketjun tilojen jakauma on  $f$



- kuinka valita siirtymätodennäköisyydet  $T(x, y)$ ?

# Milloin pitää hylätä?

- ilmaistaan käänteisyysehto seuraavasti:

$$f(x) q(x, x') \alpha(x, x') = f(x') q(x', x),$$

- $\alpha(x, x')$  on siis todennäköisyys, että kandidaatti hyväksytään
- jos  $f(x) q(x, x') > f(x') q(x', x)$  hyväksytään aina
- muuten

$$\alpha(x, x') = \frac{f(x') q(x', x)}{f(x) q(x, x')}.$$

# Hyväksymistodennäköisyys

---

- M-H -algoritmi edellyttää vain, että  $\frac{f(x')}{f(x)}$  pystytään laskemaan
- ei muita oletuksia jakaumasta
- käytännön kannalta sopivan ehdotusjakauman  $Q$  löytäminen tärkeää

# Konvergenssi

---

- alkutila vaikuttaa tilojen jakaumaan
- jotta vaikutus eliminoitaisiin tarvitaan “lämmittelyjakso”, jonka aikana ei vielä kerätä generoituja arvoja
- mikä on sopiva lämmittelyjakson pituus?
  - vaikea ongelma
- toinen keskeinen ongelma on, kuinka kauan simulaatiota pitää jatkaa, jotta otos olisi kattava

# Yhteenveto

---

- satunnaislukujen generoiminen halutusta jakaumasta on tarpeen esimerkiksi Monte Carlo -testauksen yhteydessä
- tietokoneen generoimat satunnaisluvut ovat pseudo-satunnaislukuja
- tasaisesti jakautuneiden satunnaislukujen avulla voidaan generoida muita jakaumia noudattavia lukuja
  - transformaatiomenetelmä käyttää halutun jakauman kertymäfunktion käänteisfunktiota
  - hylkäämismenetelmässä käytetään apufunktiota josta osataan generoida otoksia ja joka “sulkee sisäänsä” halutun tiheysfunktion

# Otokset tietokannoista



# Lähteet

---

- Jeffrey Scott Vitter: Faster Methods for Random Sampling, *Communications of ACM* 27, 7, 703–718, 1984.
- Numerical Recipes in C, luku 7.3

# Luennon runko

---

- käydään läpi neljä algoritmia joilla voidaan tehdä otantaa tietokannoista
- yksinkertaisimmasta tehokkaimpaan
- kaksi algoritmeista hyödyntää aiemmin käsiteltyjä transformaatio- ja hylkäämismenetelmiä
- näin saadaan kuvaa näistä menetelmistä myös realistisemmassa ympäristössä kuin viime luennon esimerkeissä



# Johdanto

---

- usein tarpeellinen tehtävä on valita  $n < N$  tietuetta satunnaisesti tietokannasta, jossa  $N$  tietuetta
- talletettuna toissijaiseen muistiin hitaalle välineelle
- kriittinen seikka on kuinka hyvin voidaan välttää turhat luku/kirjoitusoperaatiot
- halutaan saada tuloksia, jotka pätevät koko aineistoon, joutumatta käymään läpi koko aineistoa
- ei samaa tietuetta kahdesti
- myös tietuekoko voi olla suuri
- esim. laaduntarkkailu, ostoskorianalyysi

# Perusmenetelmä P

---

1. **do** {
  2. generoi satunnainen kokonaisluku  $1 \leq k \leq N$
  3. valitse  $k$ :s tietue, ellei sitä ole jo valittu
4. **}** **until**  $n$  tietuetta valittu

- 
- vaatii  $O(n)$  satunnaisluvun generoimista
  - generointiaika  $O(n)$ , mikäli 3. askelen vaatima tarkistus tapahtuu vakioajassa
  - tämä vaatii tilaa  $O(N)$  jos tarkistukseen käytetään binääristä taulukkoa, tai  $O(n)$  osoitinta hajauttamalla
  - $n$  (levy)hakua, mahdollisesti useita hakuja samaan levylohkoon tms.

# Sijaintijärjestys

---

- algoritmi valitsee tietueet ottamatta huomioon niiden sijaintijärjestystä tiedostossa
- esim. 84. tietue voi tulla valituksi ennen 16. tietuetta
- jos otoksen tietueet halutaan poimia järjestyksessä (esim. yhdellä tietokannan läpikäynnillä), tietuenumerot täytyy lajitella ennen poimintaa
- lajitteluaika  $O(n \log n)$ , esim. quicksort, heapsort
- poimintaa ei voida aloittaa, ennen kuin kaikki tietuenumerot on generoitu
- myöskään tulostusta ei voi tehdä on-line

# Otanta järjestyksessä

---

- generoidaan otokseen tulevien tietueiden numerot suoraan järjestyksessä (on-line)
- ratkaisuperiaate:
- kun  $N$  tietueesta on valittava  $n$  kpl, kunkin yksittäisen tietueen valinnan todennäköisyys on  $n/N$ 
  - $N$  on vielä käsittelemättömien tietueiden määrä
  - $n$  on otokseen vielä tarvittavien tietueiden määrä
  - jokaisen tietueen kohdalla valinta todennäköisyydellä  $n/N$
  - kun  $n = 0$ , algoritmi päättyy

# Algoritmi S

---

1. **do** {
  2. generoi satunnaisluku  $u \sim Tas(0, 1)$
  3. **if**  $u \leq n/N$  valitse seuraava tietue otokseen ja aseta  
 $n \leftarrow n - 1$
  4. **else** ohita seuraava tietue
  5. aseta  $N \leftarrow N - 1$
6. **}** **while**  $n > 0$

# Algoritmin S analyysia

---

- jos jossakin vaiheessa  $n = N$  kaikki jäljellä olevat tietueet valitaan otokseen
- koska jokaisen tietueen kohdalla on tehtävä päätös, otetaanko se mukaan vai ei  $\rightarrow O(N)$ -aikavaativuus

# Etäisyyksien generointi

- nopeammissa algoritmeissa generoidaan suoraan peräkkäisten mukaan otettavien tietueiden etäisyyksiä:
  - merkitään  $S(n, N)$ :llä satunnaismuuttujaa, joka kertoo, kuinka monta seuraavaa tietuetta hylätään
  - ratkaisuperiaate: toistetaan seuraavia askelia, kunnes  $n = 0$ 
    1. generoidaan satunnaisluku  $S(n, N)$
    2. hylätään seuraavat  $S(n, N)$  tietuetta ja valitaan niitä seuraava
    3.  $N \leftarrow N - S(n, N) - 1, n \leftarrow n - 1$



# Satunnaismuuttuja $S(n, N)$

- diskreetti satunnaismuuttuja, joka saa arvoja väliltä  $0, \dots, N - n$
- merkitään seuraavassa lyhyesti  $S$
- pistetodennäköisyysfunktio  $P(S = s) = f(s)$  ilmaisee, mikä on todennäköisyys, että sivuutetaan täsmälleen  $s$  arvoa
- jakaumafunktio  $P(S(n, N) \leq s) = \sum_{i=0}^s f(i) = F(s)$  ilmaisee, mikä on todennäköisyys että sivuutetaan enintään  $s$  arvoa

- $f(0) = n/N$
- $f(s) =$   
 $P(\text{sivuutetaan yli } s - 1 \text{ arvoa}) \cdot P(\text{valitaan } s + 1:s \text{ arvo} \mid$   
 $\text{sivuutetaan yli } s - 1 \text{ arvoa}) = (1 - F(s - 1)) \cdot n/(N - s)$
- $f$ :n ja  $F$ :n suljetut muodot hankalia laskea
- $$f(s) = \frac{n(N - n)!(N - s - 1)!}{N!(N - n - s)!}$$
- huom: laskeminen paljon tehokkaampaa kun kertomien osamääriä on ensin sievennetty

# Transformaatiomenetelmä

- seuraava algoritmi A perustuu transformaatiomenetelmään (sovellettuna diskreettiin jakaumaan)
- tavoitteena generoida satunnaismuuttuja  $s = S(n, N)$  jakaumasta  $F$  (tai paremminkin  $F(n, N)$ )
- apumuuttuja  $u \sim \text{Uas}(0, 1)$
- käänteisfunktiota  $F^{-1}(u)$  ei tunneta
- sen sijaan tarkastellaan  $F(s)$ :n arvoja
- valitaan  $s$  jolla  $F(s - 1) < u \leq F(s)$
- $F(i)$ :n ja  $f(i)$ :n arvot lasketaan tehokkaasti edellisistä arvoista  $F(i - 1)$  ja  $f(i - 1)$

# Algoritmi A

---

1. **do** {
  2. generoi satunnaisluku  $u \sim Tas(0, 1)$
  3.  $s \leftarrow 0; F(0) \leftarrow n/N$
  4. **while**  $u > F(s)$  sivuuta tietue:
    5.  $s \leftarrow s + 1; f(s) \leftarrow (1 - F(s - 1))n/(N - s);$   
 $F(s) \leftarrow F(s - 1) + f(s)$
  6. valitse  $s + 1$ :s tietue
  7. aseta  $n \leftarrow n - 1, N \leftarrow N - s - 1$
8. **}** **while**  $n > 0$

# Algoritmin A analyysiä

---

- do-silmukka suoritetaan  $n$  kertaa
- while-silmukka (askel 5) vie kaikkiaan aikaa luokkaa  $O(N)$ , koska jokainen tietue käydään läpi kerran
- algoritmin aikavaativuus on siis  $O(N)$  (kuten Algoritmi S)
- generoitavien satunnaislukujen määrä on  $n$  eli huomattavasti pienempi

# Hylkäämismenetelmä

- seuraavassa algoritmissa  $D$  käytetään satunnaismuuttuja  $S(n, N)$ :n generointiin hylkäämismenetelmää (sekä transformaatiomenetelmää apujakaumalle  $g$ )
- apujakauma  $g$  on jatkuva
  - muunnetaan diskreetti pistetodennäköisyysfunktio jatkuvaksi porraskäyräksi
  - portaan pituus on 1 ja funktion arvo  $i$ :nnellä portaalla on pistetodennäköisyys  $p(i - 1)$
  - ks. kuva Numerical Recipes, s. 294
- hylkäysmenetelmässä  $s$ :n arvoksi käytetään vain kokonaislukuosaa  $\lfloor x \rfloor$

- 
- tehtävä siis: generoi satunnaislukuja  $S$ :n jakaumasta
  - olkoon  $g$  sopiva apujakauma, joka approksimoi jakaumaa  $F(s)$  “hyvin”
  - etsitään  $c \geq 1$  jolle  $f(\lfloor x \rfloor) \leq cg(x)$  kaikilla  $x$  relevantilla arvoalueella
  - generoidaan jakaumasta  $g$  kandidaatti  $x$
  - generoidaan satunnaisluku  $u$  väliltä  $0, \dots, 1$
  - jos  $u > f(\lfloor x \rfloor)/(cg(x))$  hylätään  $\lfloor x \rfloor$
  - muuten hyväksytään  $s \leftarrow x$

# Litistys (*squeeze method*)

---

- ehdon  $u > f(\lfloor X \rfloor) / (cg(X))$  testaaminen vie jonkin verran aikaa, koska  $f$  on työläs laskea
  - hylkäämisen todennäköisyys on pieni
- ⇒ aikaa voidaan säästää korvaamalla  $f(s)$  funktiolla  $h(s)$  joka voidaan laskea nopeammin ja jolle  $h(s) \leq f(s)$
- $f(\lfloor X \rfloor)$  täytyy laskea vain kun  $u > h(\lfloor X \rfloor) / cg(X)$
  - litistystekniikka:  $h(\lfloor X \rfloor) \leq f(\lfloor X \rfloor) \leq cg(X)$
  - litistys ei tietysti ole välttämätöntä algoritmin toiminnan kannalta



# Algoritmi D

---

1. **do** {
  2. generoi satunnaisluku  $u \sim Tas(0, 1)$
  3. generoi satunnaisluku  $x$  jakaumasta  $g(x)$
  4. **if**  $u \leq h(\lfloor x \rfloor) / (cg(x))$  aseta  $s \leftarrow \lfloor x \rfloor$
  5. **else if**  $u \leq f(\lfloor x \rfloor) / (cg(x))$  aseta  $s \leftarrow \lfloor x \rfloor$
  6. **else goto 2**
  7. sivuuta  $s$  tietuetta ja valitse sen jälkeen seuraava otokseen
  8. aseta  $n \leftarrow n - 1, N \leftarrow N - s - 1$
9. **}** **while**  $n > 0$

# Sopivan funktion $g$ valinta

---

- mikä on sopiva apufunktio  $g$ ?
- $S$ :n arvo on pienin  $n$ :stä *kokonaisluvusta*, jotka on generoitu (“ilman takaisinpanoa”) toisistaan riippumattomasti väliltä  $0, \dots, N$
- generoidaan otos “vastaavasta” jatkuvasta jakaumasta (“takaisinpanolla”)
- olkoon  $X$  satunnaismuuttuja ko. jakaumasta
- $X$ :n arvo voidaan ajatella pienimmäksi  $n$ :stä riippumattomasta *reaaliluvusta*  $\text{Tas}(0, N)$ -jakaumasta

# Kandidaattien generointi

- kandidaatin  $x$  generoiminen jakaumasta  $g$  on helppoa ja nopeaa
- käytetään transformaatiomenetelmää
- merkitään  $Z_1, Z_2, \dots, Z_n$  riippumattomia  $\text{Tas}(0, N)$ -jakautuneita reaalitylukuja
- näitä lukuja ei suinkaan tarvitse generoida, niitä käytetään tässä vain  $g$ :tä vastaavan jakaumafunktion  $G(x)$  johtamiseen

$$\begin{aligned} G(x) &= P(X \leq x) = 1 - P(X > x) \\ &= 1 - \prod_{1 \leq k \leq n} P(Z_k > x) = 1 - \left(1 - \frac{x}{N}\right)^n. \end{aligned}$$

# Kandidaattien generointi (2)

- haluttu  $x$  saadaan siis transformaatiomenetelmällä  
 $x = G^{-1}(u)$
- edellyttää tietysti että sellainen on (tässä sivuutetaan sen näyttäminen intuitioon vedoten, samoin tekninen käänteisfunktion johtaminen)
- saadaan  $G^{-1}(u) = N(1 - (1 - u)^{1/n})$
- koska muuttuja  $1 - U$  on tasaisesti jakautunut kun  $U$ :kin on, saadaan kandidaatti  $x$  asettamalla

$$x \leftarrow N(1 - u^{1/n})$$

# Muut parametrit

- myös vakio  $c$  ja litistämiseen käytettävä funktio  $h$  valittava
- $h$ :n oltava tiukka alaraja, toisaalta sellainen että se pystytään laskemaan nopeasti

■

$$c = \frac{N}{N - n + 1}$$
$$h(s) = \frac{n}{N} \left( 1 - \frac{s}{N - n + 1} \right)^{n-1}$$

# Algoritmin analyysiä

---

- kun  $n$  on suuri suhteessa  $N$ :ään, algoritmi saattaa olla hitaampi kuin algoritmi  $A$
- do-silmukka suoritetaan  $n$  kertaa
- enää ei tarvitse käydä sen sisällä läpi tietueita vaan sivuutettavien tietueiden lukumäärä saadaan generoitua suoraan satunnaislukuna vastaavan satunnaismuuttujan jakaumasta
- algoritmin aikavaativuus on siis  $O(n)$

# Yhteenveto

---

- usein tarpeellinen tehtävä on valita otos satunnaisesti tietokannasta
- turhien luku-/kirjoitusoperaatioiden välttäminen tärkeää
- siksi tietueet on syytä valita järjestyksessä jossa ne on talletettu
- hylkäämismenetelmään perustuva algoritmi suoriutuu tehtävästä lineaarisessa ajassa suhteessa otoksen kokoon