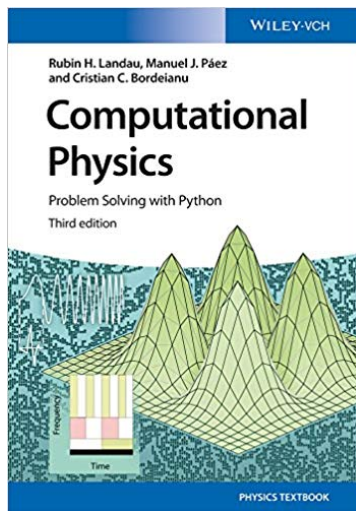
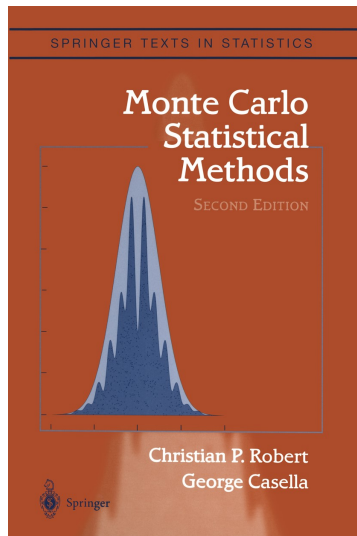


Monte Carlo módszerek

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék



Témák:

- Bevezetés: Monte Carlo integrálás alapjai
- Álvéletlen számsorozatok
- Monte Carlo integrálás
- Különböző eloszlások generálása egyenletes eloszlású számokból
- Metropolis-Hastings algoritmus
- χ^2 illesztés Monte Carlo optimalizációval
- Szimulált hőkezelés

Hogyan számoljunk bonyolult integrálokat?

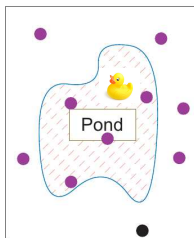


Figure: Kacsaúsztató bekerített terület közepén.

Mekkora a kacsaúsztató területe?

Hogyan számoljunk bonyolult integrálokat?

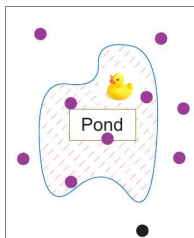


Figure: Kacsaúsztató bekerített terület közepén.

Mekkora a kacsaúsztató területe?

Ha véletlenszerűen és egyenletesen dobálunk be kavicsokat a kerítésen belüli területre, a keresett terület:

$$A_{pond} = \frac{N_{pond}}{N_{össz}} A_{negyzet} = \frac{N_{pond}}{N_{pond} + N_{negyzet}} A_{negyzet}$$

Formálisabban

Az $f(\mathbf{r}) \equiv 1$ függvényt integráltuk egy bonyolult tartományon

$$A_{pond} = \iint_A f(\mathbf{r}) d\mathbf{r}$$

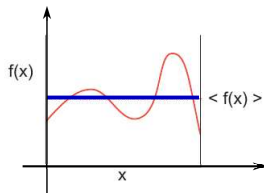
Formálisabban

Az $f(\mathbf{r}) \equiv 1$ függvényt integráltuk egy bonyolult tartományon

$$A_{pond} = \iint_A f(\mathbf{r}) d\mathbf{r}$$

Tovább lépés: tetszőleges függvény esetén igaz

$$\int_a^b f(x) dx = \langle f \rangle (b - a) \quad \langle f \rangle : \text{átlag}$$



Integrál: görbe alatti terület

Az integrál Monte-Carlo közelítése

Alapegyenlet:

$$\int_V f(\mathbf{r}) \, d\mathbf{r} \approx V \langle f \rangle \pm \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}$$

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^N f(\mathbf{r}_i) \quad \mathbf{r}_i \in V$$

- az integrált értékét a V integrálási tartomány térfogatának és a függvény átlagértékének szorzataként írjuk fel
- ez általában igaz, ha a függvény átlagát pontosan ismernénk
- most a függvény átlagát becsüljük
- véletlenszerűen generálunk \mathbf{r}_i pontokat a V integrálási tartományon belül
- \mathbf{x}_i választása V -ből egyenletes eloszlással!
- a hiba (egy standard eltérés) $\sim 1/\sqrt{N}$ -nel csökken

Random minta választása

A Monte Carlo integráláshoz kell:

- ismerni a V integrálási tartomány térfogatát
- random vektorokat választani V -ből

Téglatest integrálási tartomány esetén egyszerű

- a tartomány térfogatát könnyen számolhatjuk
- D darab álvéletlen számot generálunk, ahol D a dimenzió
- kell egy jó álvéletlenszám-generátor: legyen kicsi az autokorreláció

$$R_k = \langle x_i x_{i+k} \rangle \rightarrow 0$$

Random minta választása bonyolult tartományból

A Monte Carlo integráláshoz kell:

- ismerni a V integrálási tartomány térfogatát
- random vektorokat választani V -ből

Bonyolult integrálási tartomány esetén: V nem feltétlen ismert

- fedjük le a tartományt egy téglatesttel, ennek $V_{teгла}$ térfogata ismert
- generáljunk pontokat a téglatesten belül
- ellenőrizzük, hogy a pont belül esik-e az integrálási tartományon
- ez utóbbit gyorsan el kell tudni dönteni
- ha a pont belül esik az integrálási tartományon, akkor felhasznájuk az $\langle f \rangle$ átlag számításához
- az ismeretlen V térfogat a generált pontok elfogadási valószínűségével arányos:

$$V = \frac{\#elfogadott\ pontok}{\#összes\ pont} \cdot V_{teгла}$$

Álvéletlen számsorozat, egyszerű tesztek

A számítógép álvéletlen számokat tud generálni

- determinisztikus algoritmus által előállított számok
- statisztikai tulajdonságai alapján mégis véletlenszerűnek tűnik

A számítógép (ál)véletlenszám generátora általában egyenletes eloszlással, $[0, 1]$ intervallumból generál számokat

Álvéletlen számsorozat, egyszerű tesztek

A számítógép álvéletlen számokat tud generálni

- determinisztikus algoritmus által előállított számok
- statisztikai tulajdonságai alapján mégis véletlenszerűnek tűnik

A számítógép (ál)véletlenszám generátora általában egyenletes eloszlással, $[0, 1]$ intervallumból generál számokat

Ellenőrzés

- ha valóban véletlenszerű, egyenletes eloszlásúak a generált számok:

$$\int_0^1 x^k P(x) dx \approx \frac{1}{N} \sum_{i=1}^N x_i^k = \frac{1}{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

Álvéletlen számsorozat, egyszerű tesztek

A számítógép álvéletlen számokat tud generálni

- determinisztikus algoritmus által előállított számok
- statisztikai tulajdonságai alapján mégis véletlenszerűnek tűnik

A számítógép (ál)véletlenszám generátora általában egyenletes eloszlással, $[0, 1]$ intervallumból generál számokat

Ellenőrzés

- ha valóban véletlenszerű, egyenletes eloszlásúak a generált számok:

$$\int_0^1 x^k P(x) dx \approx \frac{1}{N} \sum_{i=1}^N x_i^k = \frac{1}{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

- Ha x_i és x_{i+k} közös eloszlását leíró $P(x_i, x_{i+k})$ -ra igaz, hogy x_i és x_{i+k} függetlenek, egyenletes eloszlással:

$$\int_0^1 dx \int_0^1 dy x \cdot y \cdot P(x, y) \approx \frac{1}{N} \sum_{i=1}^N x_i x_{i+k} = \frac{1}{4} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

Monte Carlo integrálás tulajdonságai

A Monte Carlo integrálás implementálása elég egyszerű, de vannak problémái

- lassan konvergál
- kis tartományon nagyokat változó függvényekkel nehezen boldogul

Miért alkalmazzák mégis?

Monte Carlo integrálás tulajdonságai

A Monte Carlo integrálás implementálása elég egyszerű, de vannak problémái

- lassan konvergál
- kis tartományon nagyokat változó függvényekkel nehezen boldogul

Miért alkalmazzák mégis?

- egydimenziós integrálok numerikus kiszámítására léteznek kifinomult módszerek
- elvileg ezek 2-3 dimenziós esetben is alkalmazhatóak, ha az integrandus és az integrálási határok nem túl bonyolultak
- gyakorlatban az integrál kiszámolásához szükséges függvénykiértékelések lehetséges maximális száma ad egy felső korlátot: N_{max}

Monte Carlo integrálás tulajdonságai

Miért alkalmazzák mégis?

- pl ha a dimenziók száma $D = 6$, és dimenzióként $N_{1D} = 10$ integrálási pontot veszünk, akkor összesen $N_{max} = (N_{1D})^D = 10^6$ kiértékelés
- a D dim integrált D darab 1 dim integrál segítségével számoljuk
- az egyes 1 dim integrálok hibája $\mathcal{O}([\frac{1}{N_{1D}}]^4) \sim 10^{-4}$
- a D dim integrál teljes hibája tehát $\sim D \cdot \mathcal{O}([\frac{1}{N_{1D}}]^4) \sim 6 \cdot 10^{-4}$
- az integrálási határok bonyolultságából eredő esetleges problémákat még is vettük figyelembe
- a Monte Carlo integrálás hibája $\sim \frac{1}{\sqrt{N_{max}}} \sim 10^{-3}$ függetlenül D -től
 \Rightarrow ugyanaz a nagyságrend

Sokdimenziós integrálokat gyakorlatilag csak Monte Carlo módszerrel lehet elvégezni!

Fejlettebb Monte Carlo integrálási módszer(ek)

Ilyenekből több is van, itt most a következő kérdést tekintjük:

Az integrandus gyakran olyan, hogy kis tartományon nagyot változik, pl:

$$\int_{-4}^4 e^{-x^2/2} dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$

Ha $[-4, 4]$ intervallumon egyenletes valószínűséggel választjuk az integrálási pontokat, akkor sok x_i esetén $f(x_i) \approx 0 \Rightarrow$ sok függvénykiértékelések alig ad járulékot

Fejlettebb Monte Carlo integrálási módszer(ek)

Ilyenekből több is van, itt most a következő kérdést tekintjük:

Az integrandus gyakran olyan, hogy kis tartományon nagyot változik, pl:

$$\int_{-4}^4 e^{-x^2/2} dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$

Ha $[-4, 4]$ intervallumon egyenletes valószínűséggel választjuk az integrálási pontokat, akkor sok x_i esetén $f(x_i) \approx 0 \Rightarrow$ sok függvénykiértékelések alig ad járulékot

Válasszunk x_i -t a fontos tartományból! \Rightarrow **importance sampling**

Importance sampling

Formálisan: ha az \mathbf{r}_i pontok egy V térfogtból valamilyen $p(\mathbf{r})$ sűrűségfüggvény szerint választjuk, ahol

$$\int p(\mathbf{r})dV = 1$$

akkor

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

ahol

$$\left\langle \frac{f}{p} \right\rangle = \frac{1}{N} \sum_i \frac{f(\mathbf{r}_i)}{p(\mathbf{r}_i)}$$

Importance sampling

Formálisan: ha az \mathbf{r}_i pontok egy V térfogatból valamilyen $p(\mathbf{r})$ sűrűségfüggvény szerint választjuk, ahol

$$\int p(\mathbf{r})dV = 1$$

akkor

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

ahol

$$\left\langle \frac{f}{p} \right\rangle = \frac{1}{N} \sum_i \frac{f(\mathbf{r}_i)}{p(\mathbf{r}_i)}$$

- hogyan válasszuk $p(\mathbf{r})$ -t?
- hogyan generáljunk \mathbf{r}_i pontokat adott $p(\mathbf{r})$ sűrűségfüggvény szerint?
Az álvéletlenség-generátorok egyenletes eloszlással generálnak számokat.

Importance sampling

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

- hogyan válasszuk meg a p -t?

Importance sampling

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

- hogyan válasszuk meg a p -t?

Amennyire lehet, legyen $f(\mathbf{r})/p(\mathbf{r})$ konstans! \Rightarrow “reduction of variance”
Meg lehet mutatni, hogy $p(\mathbf{r}) \sim |f(\mathbf{r})|$ (lásd Numerical Recipes, Eq.(7.8.4)-(7.8.7))

Importance sampling

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

- hogyan válasszuk meg a p -t?

Amennyire lehet, legyen $f(\mathbf{r})/p(\mathbf{r})$ konstans! \Rightarrow “reduction of variance”
Meg lehet mutatni, hogy $p(\mathbf{r}) \sim |f(\mathbf{r})|$ (lásd Numerical Recipes, Eq.(7.8.4)-(7.8.7))

- hogyan generáljunk valamilyen $p(\mathbf{r})$ eloszlás szerint véletlen számokat? Önmagában is érdekes probléma!

Importance sampling

$$\int f(\mathbf{r})dV = \int \frac{f(\mathbf{r})}{p(\mathbf{r})}p(\mathbf{r})dV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2 \rangle - \langle f/p \rangle^2}{N}}$$

- hogyan válasszuk meg a p -t?

Amennyire lehet, legyen $f(\mathbf{r})/p(\mathbf{r})$ konstans! \Rightarrow “reduction of variance”
Meg lehet mutatni, hogy $p(\mathbf{r}) \sim |f(\mathbf{r})|$ (lásd Numerical Recipes, Eq.(7.8.4)-(7.8.7))

- hogyan generáljunk valamilyen $p(\mathbf{r})$ eloszlás szerint véletlen számokat? Önmagában is érdekes probléma!

Tekintsük először az egydimenziós problémát: $p(x)$

Két fő módszer

- Ha $p(x)$ kumulált eloszlásának inverze ismert: inverzeloszlás-módszert
- Ha $p(x)$ kumulált eloszlása nem invertálható: elfogadás-elvetés módszer

Inverzelosztás-módszer

Az álvéletlenszám-generátorok egyenletes eloszlással generálnak számokat:

$$p(x)dx = \begin{cases} dx & 0 < x < 1 \\ 0 & \text{egyébként} \end{cases}$$

Mi lesz az eloszlása annak az valószínűségi változónak, amelyet egy $y(x)$ függvény segítségével kapunk?

$$|p(y)dy| = |p(x)dx| \rightarrow p(y) = p(x) \left| \frac{dx}{dy} \right|$$

Inverzelozlás-módszer

Az álvéletlenszám-generátorok egyenletes eloszlással generálnak számokat:

$$p(x)dx = \begin{cases} dx & 0 < x < 1 \\ 0 & \text{egyébként} \end{cases}$$

Mi lesz az eloszlása annak az valószínűségi változónak, amelyet egy $y(x)$ függvény segítségével kapunk?

$$|p(y)dy| = |p(x)dx| \rightarrow p(y) = p(x) \left| \frac{dx}{dy} \right|$$

Ha x egyenletes eloszlású és azt akarjuk, hogy $y(x)$ sűrűségfüggvénye adott $f(y)$ legyen, akkor a $\frac{dx}{dy} = f(y)$ differenciálegyenletet kell megoldani.

$$x = \int^y f(y')dy' = F(y) \quad F(y) : \text{kumulált eloszlásfüggvény}$$

Inverzeloszlás-módszer

Ha az $F(y)$ függvény invertálható, a keresett $y(x)$ függvény:

$$\Rightarrow y(x) = F^{-1}(x)$$

Példa: ha exponenciális eloszlású valószínűségi változót akarunk generálni az egyenletes eloszlású segítségével, akkor $f(y) = e^{-y}$ és $y(x) = -\ln(x)$ (Ellenőrizték!)

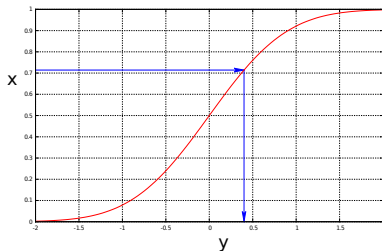
Inverzelozlás-módszer

Ha az $F(y)$ függvény invertálható, a keresett $y(x)$ függvény:

$$\Rightarrow y(x) = F^{-1}(x)$$

Példa: ha exponenciális eloszlású valószínűségi változót akarunk generálni az egyenletes eloszlású segítségével, akkor $f(y) = e^{-y}$ és $y(x) = -\ln(x)$ (Ellenőrizzék!)

Az inverzelozlás módszer geometriai interpretációja:



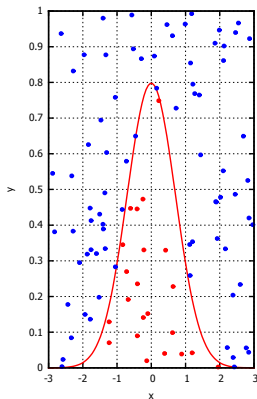
$$x = \int^y f(y') dy' = F(y)$$

- generáljunk x -t egyenletes eloszlással
- keressük meg a hozzá tartozó y értéket

Elfogadás-elvetés módszer (von Neumann módszer)

Nem szükséges, hogy a kumulatív eloszlásfüggvényt és az inverzét explicit számolni tudjuk

- húzunk egy x véletlenszámot $p(x)$ értelmezési tartományából
- húzunk egy y véletlenszámot 0 és W_0 között, ahol $W_0 > p(x), \forall x$
- mindkettőt egyenletes eloszlással
- ha $y < p(x)$, akkor megtartjuk x -et
- különben eldobjuk x -et, és újra húzunk
- a legtöbb megtartott x érték onnan jön, ahol $p(x)$ értéke a legnagyobb: x eloszlása $p(x)$ lesz
- elvileg tetszőleges D dimenzióban is működik



Az elfogadás-elvetés módszer problémái

Ha az eloszlásfüggvény nagyon “éles”

- pl. Gauss nagyon kis σ -val
- a húzott számokat majdnem mindig el fogjuk dobni

Egy dimenzióban

- léteznek ún. adaptív elfogadás-elvetés módszerek
- a $p(x)$ eloszlást valami invertálható függvénnyel majoráljuk

Több dimenzióban:

- a minta elutasításának valószínűsége exponenciálisan növekszik a dimenziószámmal \Rightarrow “**the curse of dimensionality**”
- általában is $p(\mathbf{x})$ direkt mintavételezése bonyolult lehet
- szokásos megoldás: Metropolis–Hastings-algoritmus

A Metropolis–Hastings algoritmus

“...But realize that this is the pivotal chapter of the book, one that addresses the methods that radically changed our perception of simulation and opened countless new avenues of research and applications. It is thus worth reading this chapter more than once!”

C. P. Robert and G. Casella, *Monte Carlo Statistical methods*

A Metropolis–Hastings algoritmus

“...But realize that this is the pivotal chapter of the book, one that addresses the methods that radically changed our perception of simulation and opened countless new avenues of research and applications. It is thus worth reading this chapter more than once!”

C. P. Robert and G. Casella, *Monte Carlo Statistical methods*

A legelső cikk:

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines". *Journal of Chemical Physics*. 21 (6): 1087–1092 (1953).

A Metropolis–Hastings algoritmus

Módszer egy sok elemű random minta generálására tetszőleges $p(\mathbf{x})$, többváltozós eloszlásfüggvény alapján

- $p(\mathbf{x})$ nem feltétlen normált, de integrálható
- $p(\mathbf{x})$ tetszőleges \mathbf{x} -re kiszámolható

Ez egy ún. Markov-lánc Monte Carlo módszer

- egy tetszőleges \mathbf{x}_0 random vektorból indulunk
- generálunk egy egy $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ sorozatot
- a minta $N \rightarrow \infty$ esetben megközelíti a $p(\mathbf{x})$ eloszlást

A Metropolis–Hastings-algoritmus

Kiindulás:

- tetszőleges \mathbf{x}_0 random vektorból, melyre $p(\mathbf{x}_0)$ értelmezett

Iterációs lépés:

- generálunk egy \mathbf{r} random vektort
- pl. a \mathbf{x}_i elemei körül Gauss eloszlással válsztunk értékeket
- tekintjük az $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{r}$ vektort
- ha $p(\mathbf{x}_{i+1}) \geq p(\mathbf{x}_i)$: a lépést elfogadjuk és \mathbf{x}_{i+1} része lesz a mintának
- ha $p(\mathbf{x}_{i+1}) < p(\mathbf{x}_i)$: a lépést $\frac{p(\mathbf{x}_{i+1})}{p(\mathbf{x}_i)}$ valószínűséggel fogadjuk el

Mit jelent ez?

A Metropolis–Hastings-algoritmus

Kiindulás:

- tetszőleges \mathbf{x}_0 random vektorból, melyre $p(\mathbf{x}_0)$ értelmezett

Iterációs lépés:

- generálunk egy \mathbf{r} random vektort
- pl. a \mathbf{x}_i elemei körül Gauss eloszlással válsztunk értékeket
- tekintjük az $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{r}$ vektort
- ha $p(\mathbf{x}_{i+1}) \geq p(\mathbf{x}_i)$: a lépést elfogadjuk és \mathbf{x}_{i+1} része lesz a mintának
- ha $p(\mathbf{x}_{i+1}) < p(\mathbf{x}_i)$: a lépést $\frac{p(\mathbf{x}_{i+1})}{p(\mathbf{x}_i)}$ valószínűséggel fogadjuk el

Mit jelent ez?

- ha nagyobb valószínűségű \mathbf{x}_{i+1} pontba mozdulunk, mindig elfogadjuk a lépést
- ha kisebb valószínűségű \mathbf{x}_{i+1} -be mozdulnánk, akkor valamilyen valószínűséggel elfogadjuk illetve elvetjük
- tehát általában a $p(\mathbf{x})$ nagy sűrűségű régiójában vándorol \mathbf{x}_i , de azért kisebb sűrűségű $p(\mathbf{x})$ -nek megfelelő \mathbf{x}_i -ik is megjelennek

Példa: χ^2 illesztés Monte Carlo optimalizációval

Mit lehet tenni, ha az χ^2 függvényillesztés nem vezet lineáris problémára?

- nemlineáris egyenletrendszer megoldás
- χ^2 direkt numerikus minimalizálása
- most: Monte Carlo módszer

Példa: χ^2 illesztés Monte Carlo optimalizációval

Mit lehet tenni, ha az χ^2 függvényillesztés nem vezet lineáris problémára?

- nemlineáris egyenletrendszer megoldás
- χ^2 direkt numerikus minimalizálása
- most: Monte Carlo módszer

- általános esetben a χ^2 az illesztendő paraméterek függvénye:
$$\chi^2 = \chi^2(\mathbf{a})$$
- tekintsük $\exp(-\chi^2(\mathbf{a}))$ -t egy (normálatlan) eloszlásnak \Rightarrow ez a maximum likelihood becslésből jön
- generáljunk egy random \mathbf{a}_i mintasorozatot $\exp(-\chi^2(\mathbf{a}))$ szerint
- használjuk a Metropolis-Hastings algoritmust
- milyen \mathbf{a}_{max} -ra lesz a valószínűség a legnagyobb? Konfidencia intervallumok?

χ^2 illesztés Monte Carlo optimalizációval

Probléma:

- egyenes illesztés egy olyan adathalmazra, amely esetén nem vagyunk biztosak abban, hogy mekkora az adatok σ_i szórása
- ha σ_i ismert, akkor láttuk: legkisebb négyzetek módszere

χ^2 illesztés Monte Carlo optimalizációval

Probléma:

- egyenes illesztés egy olyan adathalmazra, amely esetén nem vagyunk biztosak abban, hogy mekkora az adatok σ_i szórása
- ha σ_i ismert, akkor láttuk: legkisebb négyzetek módszere

Maximum likelihood becslés:

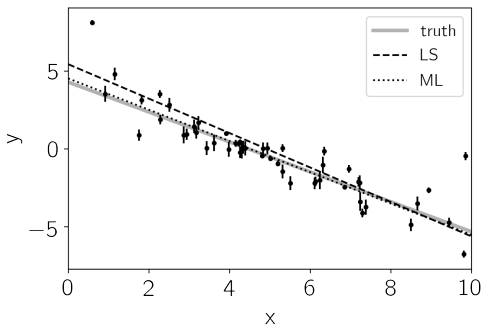
$$\ln P(y|x, \sigma, m, b, f) = -\frac{1}{2} \sum_i \left[\frac{(y_i - mx_i - b)^2}{s_i^2} \right] + \ln[2\pi s_i^2]$$

ahol

$$s_i^2 = \sigma_i^2 + f^2(mx_i + b)^2$$

- azt gondoljuk, hogy az "igazi" szórásnégyzet nagyobb σ_i^2 -nél és ezt egy f paraméterrel jellemezhetjük
- nemlineáris probléma, m , b és f paramétereket kell meghatározni

χ^2 illesztés Monte Carlo optimalizációval



- Generáltunk egy adathalmazt adott σ_i , m , b és f paraméterekkel
- tehát ismerjük az “igazi” egyenest (truth)
- megpróbáljuk a legkisebb négyzetek illesztést, adott σ_i -vel (LS)
- maximum likelihood becsléssel kapott eredmény (ML)

Részletes tárgyalás:

<https://emcee.readthedocs.io/en/v3.0.2/tutorials/line/>

Példa: Monte Carlo optimalizáció szimulált hőkezeléssel

Korábban láttuk, hogy vannak nehéz számítási problémák, pl:

- Ising modell: melyik a minimális energiájú spinkonfiguráció
- utazó ügynök: melyik az a bejárás, amelyik a legrövidebb úthosszra vezet

A lehetséges konfigurációk száma nagyon nagy

- Ising-modellnél spinek beállása: 2^N
- utazó ügynöknél az utak száma: $N!$
- tehát a konfigurációk száma exponenciálisan növekszik
- ezért nem lehet minden konfigurációt végigpróbálgatni, hogy megtaláljuk a költségfüggvény minimumot
- viszont egy adott konfigurációhoz tartozó “energia”, “úthossz” polinomiális idő alatt kiértékelhető

Monte Carlo optimalizáció szimulált hőkezeléssel

Nem feltétlenül a legjobb megoldást keressük:

- ehhez végig kellene próbálni minden konfigurációt
- megelégszünk egy majdnem minimális energiával
- ez úgysem tér el nagyon a valódi minimumtól

Monte Carlo optimalizáció szimulált hőkezeléssel

Nem feltétlenül a legjobb megoldást keressük:

- ehhez végig kellene próbálni minden konfigurációt
- megelégszünk egy majdnem minimális energiával
- ez úgysem tér el nagyon a valódi minimumtól

A rendszer tetszőleges \mathbf{x}_0 állapotából indulunk ki

- tetszőleges spin-konfiguráció
- a városok tetszőleges összekötése, stb.

Ehhez tartozik egy $E(\mathbf{x}_0)$ költségfüggvény (energia)

Monte Carlo optimalizáció szimulált hőkezeléssel

Nem feltétlenül a legjobb megoldást keressük:

- ehhez végig kellene próbálni minden konfigurációt
- megelégszünk egy majdnem minimális energiával
- ez úgysem tér el nagyon a valódi minimumtól

A rendszer tetszőleges \mathbf{x}_0 állapotából indulunk ki

- tetszőleges spin-konfiguráció
- a városok tetszőleges összekötése, stb.

Ehhez tartozik egy $E(\mathbf{x}_0)$ költségfüggvény (energia)

Definiálunk egy elemi lépést, pl:

- egy spin átfordítása
- két város sorrendjének megcserélése

Monte Carlo optimalizáció szimulált hőkezeléssel

Nem feltétlenül a legjobb megoldást keressük:

- ehhez végig kellene próbálni minden konfigurációt
- megelégszünk egy majdnem minimális energiával
- ez úgysem tér el nagyon a valódi minimumtól

A rendszer tetszőleges \mathbf{x}_0 állapotából indulunk ki

- tetszőleges spin-konfiguráció
- a városok tetszőleges összekötése, stb.

Ehhez tartozik egy $E(\mathbf{x}_0)$ költségfüggvény (energia)

Definiálunk egy elemi lépést, pl:

- egy spin átfordítása
- két város sorrendjének megcserélése

Az elemi lépés segítségével generálunk egy új *random* konfigurációt:

$$\mathbf{x}_i \rightarrow \mathbf{x}_{i+1}$$

Monte Carlo optimalizáció szimulált hőkezeléssel

Az \mathbf{x}_{i+1} lépést elfogadjuk, ha annak hatására a $E(\mathbf{x}_{i+1})$ csökken

- csökken a spin-rendszer energiája
- csökken a bejárando városok közti út

De egy kis valószínűséggel engedünk “rossz” lépést is

- különben a keresés könnyen beleragadna valamilyen lokális minimumba

Honnan vesszük a $p(\mathbf{x}_i)$, $p(\mathbf{x}_{i+1})$ valószínűségeket?

Monte Carlo optimalizáció szimulált hőkezeléssel

Az \mathbf{x}_{i+1} lépést elfogadjuk, ha annak hatására a $E(\mathbf{x}_{i+1})$ csökken

- csökken a spin-rendszer energiája
- csökken a bejárando városok közti út

De egy kis valószínűséggel engedünk “rossz” lépést is

- különben a keresés könnyen beleragadna valamilyen lokális minimumba

Honnan vesszük a $p(\mathbf{x}_i)$, $p(\mathbf{x}_{i+1})$ valószínűségeket?

PI a Boltzmann eloszlásból. Az $\mathbf{x}_i \rightarrow \mathbf{x}_{i+1}$ lépés elfogadásának valószínűsége:

$$P_{i \rightarrow i+1} = \begin{cases} 1 & \text{ha } E(\mathbf{x}_{i+1}) \leq E(\mathbf{x}_i) \\ e^{-\frac{E(\mathbf{x}_{i+1}) - E(\mathbf{x}_i)}{k_B T}} & \text{ha } E(\mathbf{x}_{i+1}) > E(\mathbf{x}_i) \end{cases}$$

Megjelent egy új paraméter: T “hőmérséklet” (k_B konstans)

Szimulált hőkezelés (simulated annealing) eljárás:

$$P_{i \rightarrow i+1} = \begin{cases} 1 & \text{ha } E(\mathbf{x}_{i+1}) \leq E(\mathbf{x}_i) \\ e^{-\frac{E(\mathbf{x}_{i+1}) - E(\mathbf{x}_i)}{k_B T}} & \text{ha } E(\mathbf{x}_{i+1}) > E(\mathbf{x}_i) \end{cases}$$

- random lépéseket generálunk
- kezdetben T nagy, ekkor nagyobb valószínűséggel fogadunk el rossz lépést
- ez azt eredményezi, hogy a keresés viszonylag szabadon ugrálhat különböző minimumok között
- valamilyen program szerint fokozatosan csökkentjük a T -t
- a keresés ráhúzódik az elérhető legjobb minimumra

Példa: Ising-lánc adott hőmérsékleten

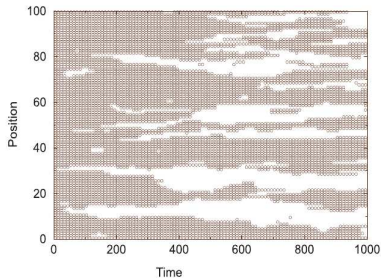


Figure: ©Computational Physics

- 100 spinből álló lánc a függőleges tengely mentén
 - a barna körök +1 spint jelölnek, a fehér mezők -1 spint
 - véges hőmérsékleten fut a szimuláció
 - a spinlánc kezdeti \mathbf{x}_0 vektora olyan állapotnak felel meg, amikor minden spin +1
 - egyesével forgatjuk át a spineket, az \mathbf{x}_i állapotvektorok az iterációs lépésekben (Time) egymás után láthatók
-
- hosszabb idő után mágneses domének láthatóak
 - a mágnesezettség véges marad (több +1 spin mint -1 spin)

Köszönöm a figyelmet!