Monte Carlo módszerek

2019. március 11.

Álvéletlen számok generálása

Álvéletlen számsorozat

- determinisztikus algoritmus által előállított számok
- statisztikai tulajdonságai alapján mégis véletlenszerűnek tűnik
- a számsort „véletlen” helyről indítjuk, pl. rendszeridő alapján inicializáljuk (seed)

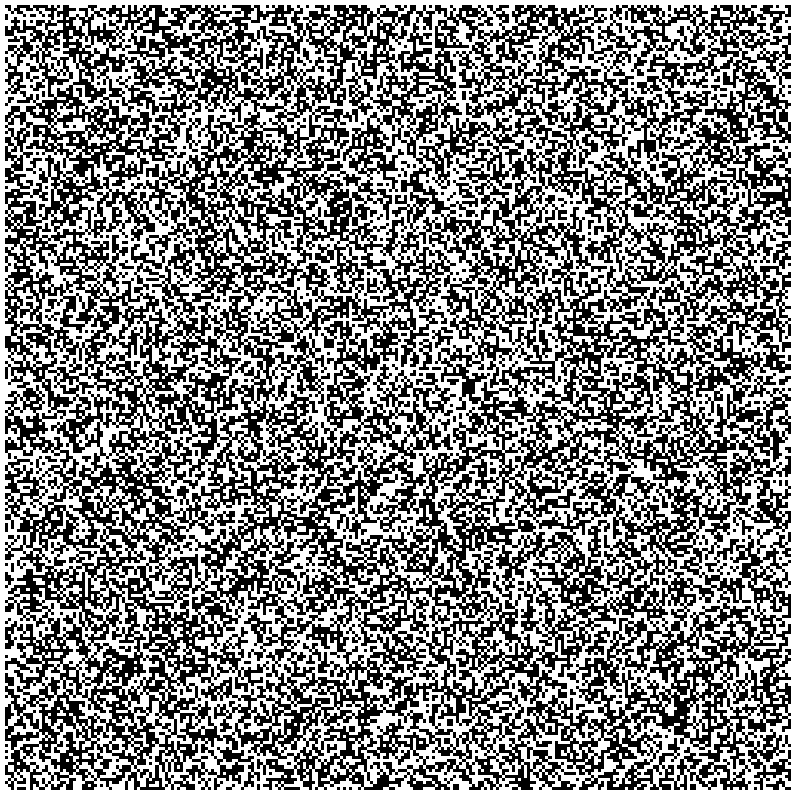
Statisztikailag véletlenszerű számsorozat

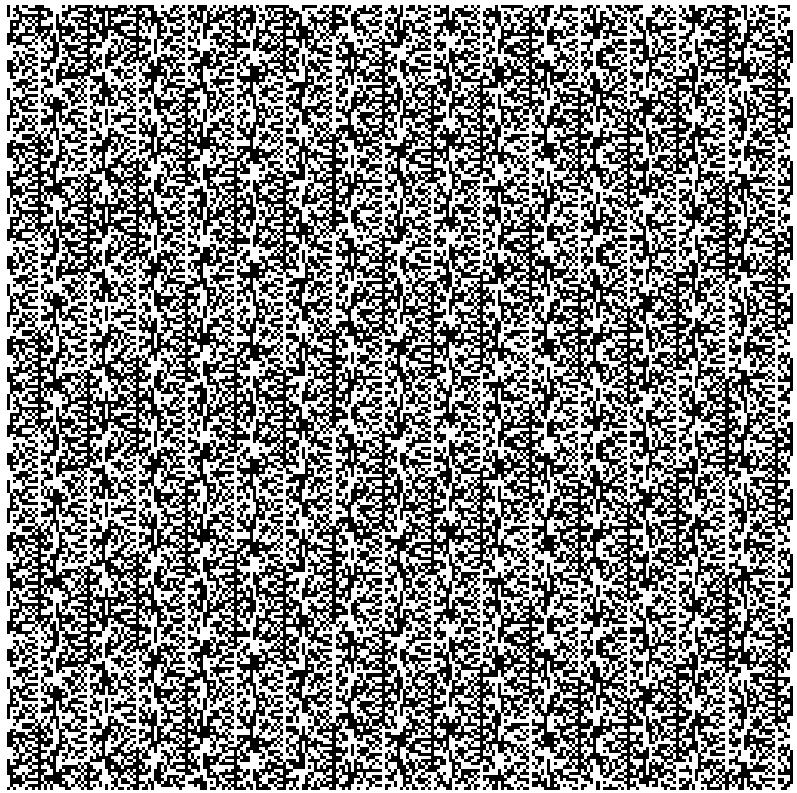
- a számsor autokorrelációja bármilyen hosszon zérus

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

- ilyet nagyon nehéz gyártani
- léteznek hardveres random szám generátorok is, de lassúak

Jó és rossz véletlen számok





Az álvéletlen számok a kriptográfiában is nagyon fontosak

- egy rossz véletlenszám generátor könnyen törhetővé teszi a módszert
- mára már hosszú lista létezik a rossz algoritmusokról

Az integrál Monte-Carlo közelítése

Alapegyenlet:

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

$$\langle f\rangle = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \qquad x_i \in V$$

Vagyis

- az integrált értékét a V integrálási tartomány térfogatának és a függvény átlagértékének szorzataként írjuk fel
- az átlagot N mintavételezéssel közelítjük
- a hiba mindössze $1/\sqrt{N}$-nel csökken

A Monte-Carlo integrálás menete

Az integrálandó f függvény értékét minden pontban ki tudjuk számolni

- véletlenszerűen generálunk egy x_i pontot a V integrálási tartományon belül
- x_i választása V-ből egyenletes eloszlással!
- meghatározzuk $f(x_i)$ értékét
- az így kapott értékeket kiátlagoljuk

Esetleg lehet nézni, hogy hogyan csökken a hiba

- egy idő után egyre kevésbé fluktuál az átlag
- ha a változás több lépés után is egy ϵ alatt van, akkor megállhatunk

Monte Carlo integrálás tulajdonságai

A Monte Carlo implementálása egyszerű, de vannak problémái

- pontatlan
- lassan konvergál
- kis tartományon nagyokat változó függvényekkel nem boldogul

Csak akkor jó, ha a következők fennállnak:

- az eredménynek nem kell nagyon pontosnak lennie
- néhány % eltérés a valódi értéktől még elfogadható
- könnyű a V tartományon belüli random x_i-ket előállítani

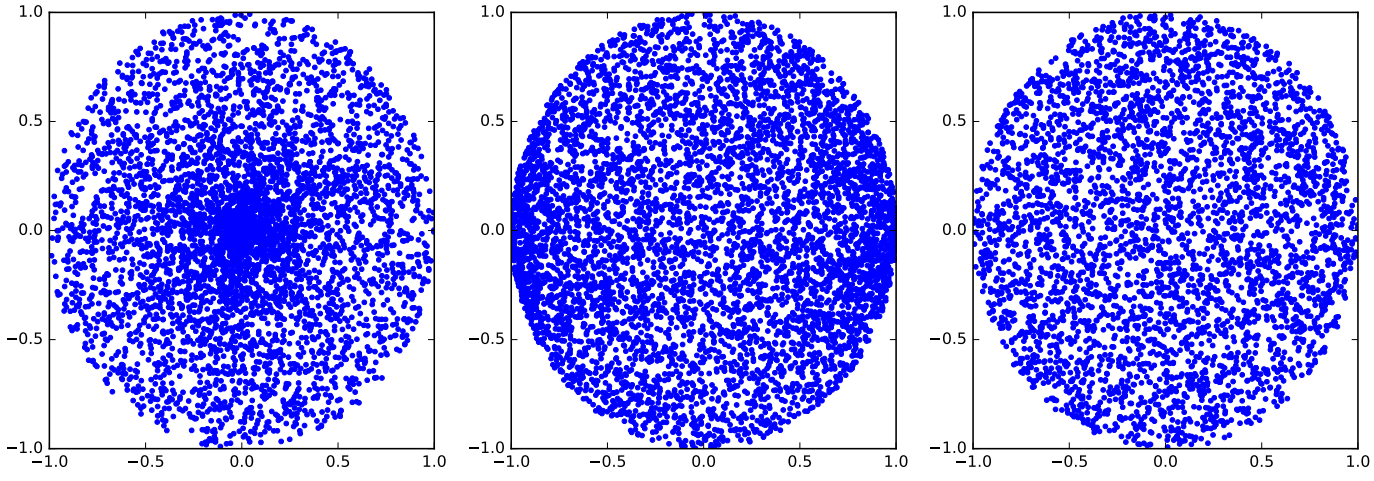
Figyelem: V-ből egyenletes eloszlással kell húzni!

Feladat: integráljuk az $f(x, y)$ függvényt az egységkörön belül.

Hogyan húzunk x és y random számokat?

- polár koordinátákat generálunk?
- random $x \in [-1, 1]$-hez olyan y-t generálunk, ami biztosan a körön belül van?

Egyik sem lesz jó, mert nem kapunk egyenletes eloszlást!



Véletlen számok adott eloszlással

Az álvéletlenszám-generátorok egyenletes eloszlással generálnak számokat

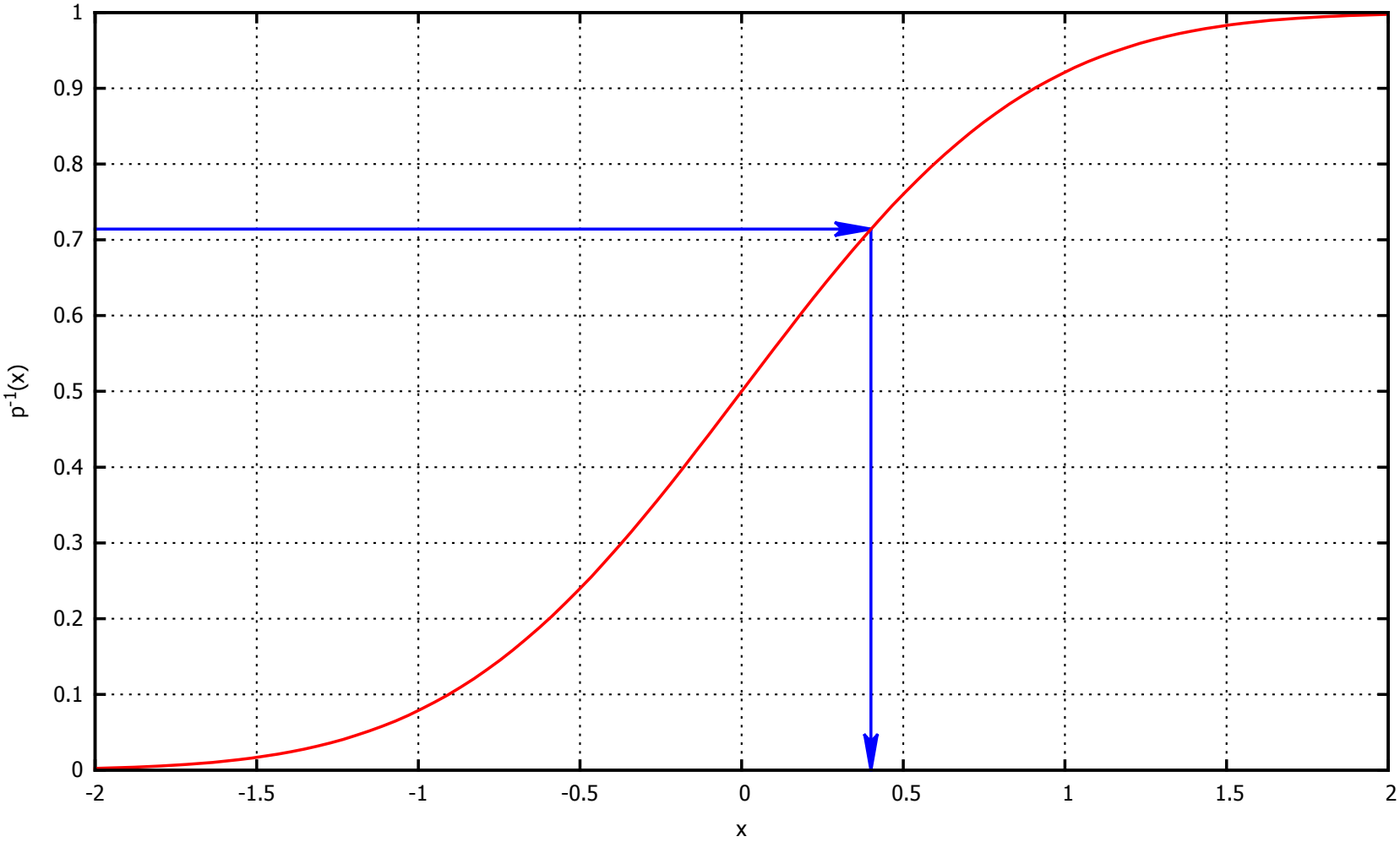
Adott egy $p(x)$ valószínűségi eloszlásfüggvény

- olyan véletlen számokat szeretnénk, mely megfelel a $p(x)$ eloszlásnak

Két fő algorimus

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

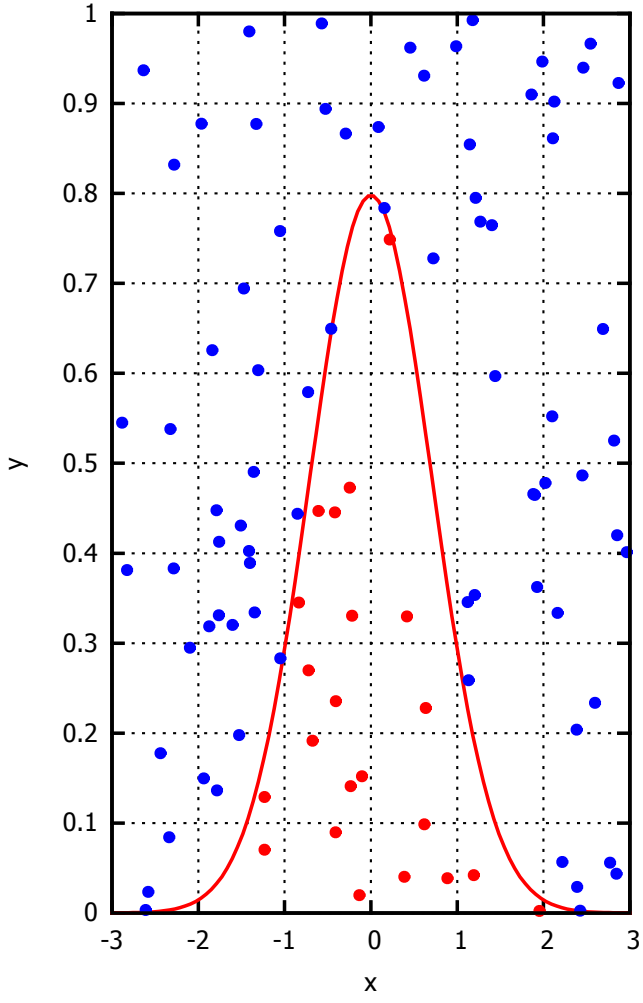
Inverzeloszlás-módszer



Elfogadás-elvetés módszer

- húzunk egy x véletlenszámot $p(x)$ értelmezési tartományából
- húzunk egy y véletlenszámot 0 és 1 között
- mindkettőt egyenletes eloszlással
- ha $y < p(x)$, akkor megtartjuk x-et
- különben eldobjuk x-et, és újra húzunk

Jé! Ez a függvény alatti terület közelítésére is jó lesz!



Random minta választása bonyolult tartományból

A Monte Carlo integráláshoz kell:

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

Téglatestek esetén ez utóbbi egyszerű

- D darab véletlen számot generálunk, ahol D a dimenzió
- csak egy jó álvéletlenszám-generátor kell hozzá
- lehetőleg nagyon kicsi legyen az álvéletlen számok autokorrelációja

Bonyolult tartomány esetén

- nem közvetlenül tartományba eső pontot generálni
- ilyenkor fedjük le a tartományt egy téglatesttel
- generáljunk pontokat a téglatesten belül
- nézzük meg, hogy a pont belül esik-e az integrálási tartományon
- ez utóbbit gyorsan el kell tudni dönteni
- ha kívül esik, akkor a pontot egyszerűen eldobjuk

További bonyodalom: V térfogata nem ismert

Ha az integrálási tartomány térfogata sem ismert

- a D dimenziós integrálási tartományt foglaljuk be egy téglatestbe
- ehhez jól meg kell becsülni a tartomány határait
- a téglatest térfogatát ismerjük: T
- generáljunk pontokat a téglatesten belül
- egy pontot elfogadunk, ha az V-ne belül esik
- az ismeretlen térfogat a pontok elfogadási valószínűségével arányos:

$$V = \frac{\#\text{elfogadott pontok}}{\#\text{összes pont}} \cdot T$$

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

- $p(x)$ direkt mintavételezése bonyolult
- szokásos megoldás: Metropolis–Hastings-algoritmus

A Metropolis–Hastings-algoritmus

Módszer egy sok elemű random minta generálására tetszőleges $p(x)$, általában többváltozós eloszlásfüggvény alapján

- $p(x)$ nem feltétlen normált, de integrálható
- $p(x)$ tetszőleges x-re kiszámolható

Ez egy ún. Markov-lánc Monte Carlo módszer

- a random minta egy $x_1, x_2, ..., x_i$ sorozatként áll elő
- a minta $N \to \infty$ esetben megközelíti a $p(x)$ eloszlást
- az x_i számsorozat autokorrelációja lehetőleg $\to 0$

Gyakori felhasználás

- minták generálása bonyolult eloszlásokból
- pl. bayesi statisztika eloszlásfüggvényei bonyolult analitikus függvények szorzataiként állnak elő

A Metropolis–Hastings-algoritmus

Kiindulás:

- tetszőleges x_0 random vektorból, melyre $p(x)$ értelmezett

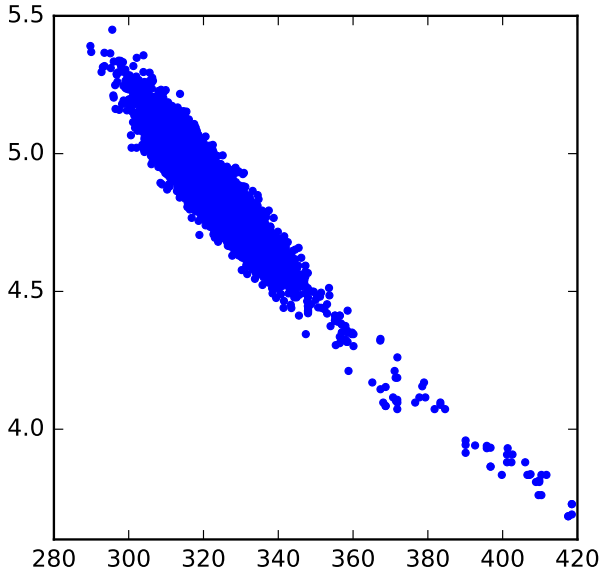
Iterációs lépés:

- generálunk egy r random vektort (D random szám)
- tekintjükaz $x_{i+1} = x_i + r$ vektort
- ha $p(x_{i+1}) \geqslant p(x_i)$:
  a lépést elfogadjuk és x_{i+1} része lesz a mintának
- ha $p(x_{i+1}) < p(x_i)$:
  a lépést $\dfrac{p(x_{i+1})}{p(x_i)}$ valószínűséggel fogadjuk el

A Metropolis–Hastings-algoritmus tulajdonságai

Burn-in

- random vektorból indulunk
- innen az algoritmus gyorsan megindul az eloszlás maximuma irányába
- viszont a mintasorozat kezdeti jópár elemét így is el kell dobni



Autokorrelációk lecsökkentése

- az autokorrelációk általában nagyobb, mert közeli pontokba lépünk
- el kell dobni a generált minta minden k. elemét

χ^2 illesztés Metropolis–Hastings-algoritmussal

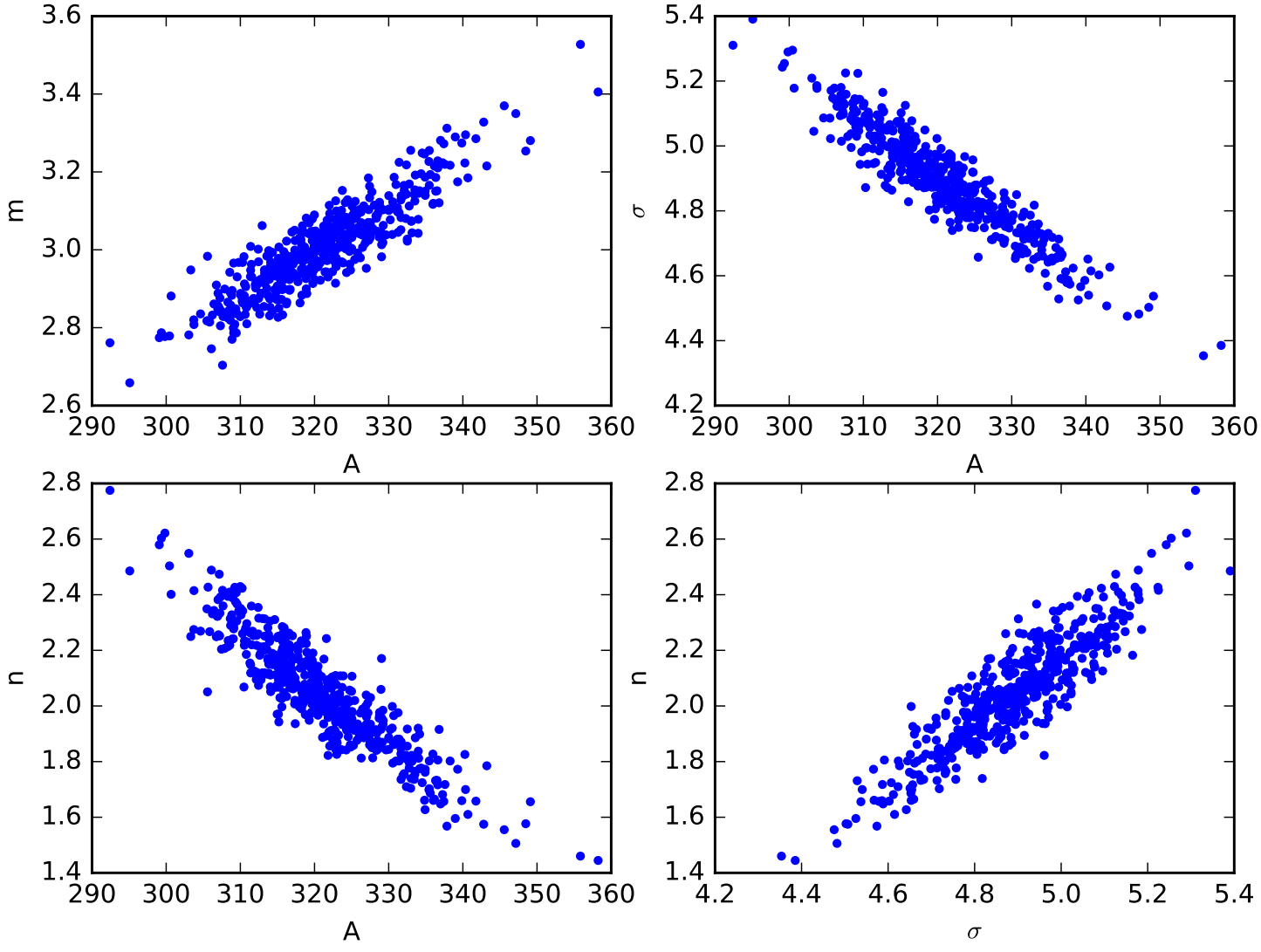
Eddig csak lineáris χ^2 illesztést néztünk

- mit lehet tenni, ha az illesztendő modell nem vezet lineáris problémára?
- általános esetben a χ^2 az illesztendő paraméterek függvénye: $\chi^2 = \chi^2(a)$
- tekintsük $\exp(-\chi^2(a))$-t egy (normálatlan) eloszlásnak
- generáljunk egy random a_i mintasorozatot $\exp(-\chi^2(a))$ szerint
- használjuk a Metropolis-Hastings algoritmust

Az így kapott a_i minta hisztogramjai jól jellemzik az illesztett paramétereket

- Gauss-eloszlás a legvalószínűbb érték körül
- Az eloszlás szórása jellemzi az illeszés hibáját
- kovarianciák is látszanak

χ^2 illesztés: paraméterek kovarianciája



χ^2 illesztés: paraméterek hisztogramja