4. sklop: Algoritem Metropolis-Hastings

Nina Ruzic Gorenjec

Imamo nek model, za katerega poznamo verjetje $L(\theta \mid x)$ imamo (θ je lahko vektor parametrov, podatki x pa matrika). Odlocimo se za apriorno porazdelitev $\pi(\theta)$, imamo njeno formulo torej poznamo.

Izracunamo torej *le se* aposteriorno porazdelitev $\pi(\theta \mid x)$ preko Bayesove formule

$$\underbrace{\pi(\theta \mid x)}_{\text{nimamo še}} \underset{\text{mujno}}{\propto} \overbrace{L(\theta \mid x)\,\pi(\theta)}^{\text{imamo}}.$$

→ To mi še gasteta! (*)
→ Manjka normalizacijska konst.
→ Integriranje ☹

Preko aposteriorne porazdelitve izracunamo *vse, kar nam srce pozeli*: za tockovno oceno vzamemo povprecje ali pa mediano, za interval zaupanja izracunamo primerne kvantile, izracunamo verjetnosti nasih domnev.

Kaj je tu problem? (*)

1 MCMC (Markov Chain Monte Carlo)

Posebno so napisani ključni izrazi (poleg angleških izrazov).

Zelimo aposteriorno porazdelitev za parameter θ, natancneje vzorec iz aposteriorne porazdelitve.

Ideja algoritma:

To je algoritem (sampler) – izberemo med že izpeljanimi.

1. Izberemo si zacetno vrednost $\theta^{(0)}$.
2. *Najdemo primerno* porazdelitev $p(\cdot|\cdot)$, tako da bomo lahko na vsakem koraku i dobili $\theta^{(i+1)}$ kot simulacijo iz porazdelitve $p(\theta^{(i+1)}|\theta^{(i)})$.
   parameter te porazdelitve npr. $p(\cdot|\theta) = N(\theta, 1)$
3. Po n korakih dobimo realizacijo $\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(n)}$. Postopek ponavljamo *dokler ne dosezemo zeljene konvergence*, pri cemer v koncnem vzorcu izpustimo zacetnih nekaj vrednosti (*burn-in*, npr. m). oz. warm-up
4. Izbrani vzorec $\theta^{(m+1)}, \theta^{(m+2)}, \ldots, \theta^{(n)}$ je vzorec iz aposteriorne porazdelitve.
5. Preizkusimo razlicne zacetne vrednosti in primerjamo dobljene rezultate.

(6. Preizkusimo razlicne apriorne porazdelitve in primerjamo dobljene rezultate.)
→ ni del preverjanja konvergence MCMC
→ Preverjamo, ali je aposteriorna stabilna glede na manjše spremembe apriorne – sensitivity analysis.

→ Preverjanje konvergence – ključno

Če veriga ne „izgleda dobro" (glej str. 3) ali verige ob različnih začetnih vrednostih ne sovpadajo, potem so razlogi za to lahko:

1. Premalo iteracij → poskusiti več (10.000, 50.000)
2. Neprimeren algoritem, tj. prepočasna konvergenca za dani model, ker ne upošteva kakšne lastnosti modela → poskusiti drug algoritem /sampler
3. Model se ne prilagaja dobro podatkom → ugotoviti zakaj, spremeniti model

2 Algoritem Metropolis-Hastings

Algoritem Metropolis-Hastings je eden izmed MCMC algoritmov.

predlagalna porazdelitev

Najprej si izberemo *proposal distribution* $q(\cdot|\cdot)$, ki je *primerne* oblike.

ključno prav izbrati

En korak Metropolis-Hastings algoritma, ko imamo iz prejsnjega koraka na voljo $\theta^{(i)}$:

1. Simuliramo kandidata θ^* iz porazdelitve $q(\cdot|\theta^{(i)})$. $\sim N(\theta^{(i)}, \sigma^2_{proposal})$
2. Izracunamo verjetnost sprejetja (*acceptance probability*) Tako v DN in spodnjem primeru; to je v splošnem tipično.

- Če θ^* ustreza podatkom, bolj kot $\theta^{(i)}$, potem bo $L(\theta^*/x)$ večje.
- Npr. za $\pi(\theta^*)$ – ob neinformativni apriorni to ni toliko pomembno.

$$\alpha = \min\left\{1, \frac{\overbrace{L(\theta^*|x)\,\pi(\theta^*)}^{\propto \pi(\theta^*|x)}\, q(\theta^{(i)}|\theta^*)}{\underbrace{L(\theta^{(i)}|x)\pi(\theta^{(i)})}_{\propto \pi(\theta^{(i)}/x)}\, q(\theta^*|\theta^{(i)})}\right\}.$$

Bistveno pri asimetrični predlagalni porazdelitvi.

Če θ^* „boljši" od $\theta^{(i)}$, bo ta kvocient velik, večji od 1, zato bo $\alpha = 1$ in zagotovo $u \le 1 = \alpha$, torej bomo θ^* sprejeli.

3. Simuliramo u iz enakomerne zvezne porazdelitve na [0,1].
4. Ce je $u \le \alpha$, izberemo kandidata (*accept*) in postavimo $\theta^{(i+1)} = \theta^*$.
   Ce je $u > \alpha$, zavrnemo kandidata (*reject*) in postavimo $\theta^{(i+1)} = \theta^{(i)}$.

V nasprotnem primeru bo $\alpha < 1$, vendar bomo θ^* vseeno morda sprejeli, in sicer v primeru $u < \alpha$, torej z verjetnostjo α. ⇝ „gamble" korak

3 Algoritem Metropolis-Hastings za primer normalnega modela z znano varianco

Uporabili bomo algoritem Metropolis-Hastings za primer iz 3. sklopa, kjer so bili nasi podatki vzorec visin (metri) studentov moskega spola:

```
x <- c(1.91, 1.94, 1.68, 1.75, 1.81, 1.83, 1.91, 1.95, 1.77, 1.98,
       1.81, 1.75, 1.89, 1.89, 1.83, 1.89, 1.99, 1.65, 1.82, 1.65,
       1.73, 1.73, 1.88, 1.81, 1.84, 1.83, 1.84, 1.72, 1.91, 1.63)
```

Ponovitev iz 3. sklopa.

Privzeli smo normalni model $N(\theta, \sigma^2 = 0.1^2)$ in zeleli oceniti θ.

Verjetje tega modela je enako

$$L(\theta \mid x) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,0.1} e^{-\frac{(x_i-\theta)^2}{2\cdot(0.1)^2}}.$$

Za apriorno porazdelitev smo si izbrali normalno porazdelitev (konjugirana v tem modelu) s povprecjem $\mu_0 = 1.78$ in varianco $\sigma_0^2 = 0.2^2$.

Za aposteriorno porazdelitev smo zato dobili normalno porazdelitev s parametroma μ_n in σ_n^2, kjer je

$$\frac{1}{\sigma_n^2} = \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2},$$

$$\mu_n = \frac{1/\sigma_0^2}{1/\sigma_0^2 + n/\sigma^2}\,\mu_0 + \frac{n/\sigma^2}{1/\sigma_0^2 + n/\sigma^2}\,\bar{x}.$$

(Zapisemo lahko tudi preko *precision*=1/varianca.)

Pravo aposteriorno porazdelitev torej poznamo. → MCMC ne potrebujemo
→ Naredimo MCMC, da lahko evalviramo konvergenco
k pravi aposteriorni – tega v praksi ne bomo mogli narediti.

Sedaj jo bomo aproksimirali s pomocjo Metropolis-Hastings algoritma.

Najprej si moramo smiselno izbrati *proposal distribution* $q(\cdot|\theta^{(i)})$. Katero bi si izbrali? Normalno s povprecjem $\theta^{(i)}$ in nekim standardnim odklonom. $\longrightarrow q_{proposal}$

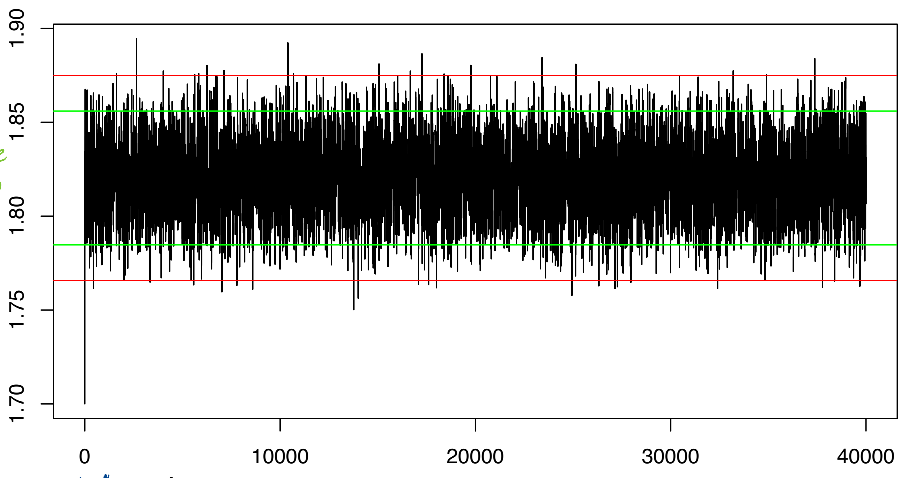
Primer zaporedja iz aposteriorne porazdelitve, ki *izgleda dobro*:

Ena veriga, chain → tipično točke med seboj povežemo

možne vrednosti

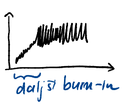
bolj pogoste vrednosti

$\theta^{(i)}$



= i, iteracija

burn-in zelo kratek tu

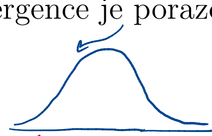


„precesavamo" porazdelitev znotraj nekega omejenega območja, večinoma smo še na nekem na ožjem „bolj verjetnem" območju.

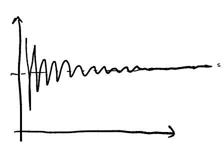
V realnem primeru ne poznamo aposteriorne in zato ne vemo, katero območje to točno je.

Na sliki smo za boljso predstavo oznacili 95% referencni interval prave aposterirone porazdelitve (zeleni crti) in odmik od povprecja aposteriorne porazdelitve za 3 standardne odklone, tj. 99.7% referencni interval (rdeci crti). **Pozor:** S tem smo uporabili vedenje o pravi aposteriorni porazdelitvi, ki v realni situaciji ni znana (ravno zato jo z MCMC metodami tudi ocenjujemo).

Pomembno: Cilj konvergence je porazdelitev, ne pa stevilo.

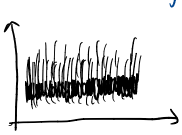


↳ NE iščemo torej tega:

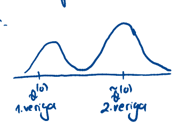


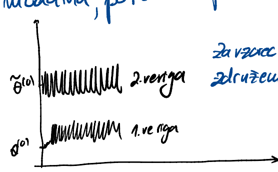
Tu je ciljna aposteriorna simetrična, zato tudi veriga simetrično raztegnjena gor in dol.

Pri asimetrični aposteriorni bi izgledala približno takole (bo pri naslednji vaji):



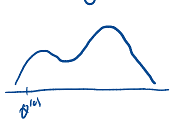
Če bi bila aposteriorna bimodalna, potem bi pri različnih začetnih vrednostih precesavali le en vrh:

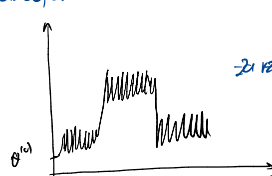




Za vzorec bi vzeli obe verigi združeni.

Oziroma bi veriga skakala med vrhoma, če ta ne bi bila tako ekstremno ločena:





Za vzorec bi vzeli vse.

Pozor, to velja ČE veste, da mora biti vaš rezultat, tj. aposteriorna, nekaj bimodalnega. Tipično to ne drži in zato so takšne slike primer slabega prilagajanja modela podatkom – preučiti razloge in spremeniti model (po tem, ko ste se prepričali, da ste vzeli dovolj dolgo verigo in da ni kriva izbira algoritma, tj. glejte str. 1).

Imamo pri tem primeru

3.1 Druga domaca naloga

3 tipe variance

fiksni

Za primer iz 3. sklopa (uporabite zgornje podatke, model z $\sigma^2 = 0.1^2$ in zgornjo apriorno porazdelitev z $\mu_0 = 1.78$ in $\sigma_0^2 = 0.2^2$ – ti parametri so fiksni) aproksimirajte aposteriorno porazdelitev s pomocjo algoritma Metropolis-Hastings, kjer sledite spodnjim korakom.

1. Sami v R-u sprogramirajte algoritem Metropolis-Hastings za primer ocenjevanja enega parametra oz. za nas primer. Kljucno je, da ga sprogramirate sami, pri cemer splosnost kode in efektivnost implementacije nista pomembni. (Za ta preprost primer boste npr. 40000 iteracij dobili v zelo kratkem casu, ne glede na izbor parametrov v spodnjih tockah ali efektivnost implementacije.) ker je model zelo preprost – če traja dolgo, ste se nekaj zmotili
2. Preizkusite ga na nasem primeru, kjer si sami izberite neko smiselno zacetno vrednost in $\boxed{\text{varianco } \textit{proposal distribution}}$. Rezultate predstavite na naslednji nacin:

   $\sigma^2_{proposal}$!

   spreminjate tekom DN
   - Narisite celotno dobljeno zaporedje $\theta^{(i)}$ (glede na iteracije i).
   - Narisite le prvih 500 ali pa 5000 clenov.
   - Narisite celotno zaporedje, kjer uporabite ustrezen *burn-in*.
   - Za tako izbrano zaporedje graficno predstavite aposteriorno porazdelitev in jo graficno primerjajte s pravo aposteriorno porazdelitvijo.
   - Ocenite parameter in 95% interval zaupanja za parameter iz izbranega zaporedja ter primerjajte z ocenami iz prave aposterirone porazdelitve.
3. Pozenite vas algoritem pri neki nesmiselni zacetni vrednosti. **Pozor:** ce boste α implementirali po formuli iz str. 2, potem algoritem pri zelo nesmiselnih zacetnih vrednostih ne bo deloval – zato je potrebno implementirati na ravni logaritma (primerno prilagodite korake algoritma). Rezultate predstavite:
   - Opisite, zakaj konkretno so se pojavile tezave, ce ste uporabili zelo nesmiselno zacetno vrednost in osnovno verzijo algoritma (brez logaritmiranja).
   - Narisite celotno dobljeno zaporedje $\theta^{(i)}$ (glede na iteracije i).
   - Narisite le prvih 500 ali pa 5000 clenov.
   - Narisite celotno zaporedje, kjer uporabite ustrezen *burn-in*.
4. Pri neki smiselni zacetni vrednosti pozenite algoritem pri nekaj razlicnih variancah za *proposal distribution*. Pri izboru pretiravajte v obe smeri (spomnite se, kaksni so po velikosti nasi podatki), tako da boste graficno opazili razlike na prvih npr. 500 iteracijah. Rezultate predstavite:
   - Za vsak primer narisite prvih nekaj (nekje med 500 in 5000) clenov in se celotno zaporedje.
   - Komentirajte razlike in zakaj do njih pride. Kaj in zakaj vas moti pri izbranih primerih?
   - **Bonus vprasanje:** Kaksen bi bil v splosnem (ne vezano na nas vzorec) vas predlog glede izbora variance *proposal distribution* oz. kaksen bi bil predlog za izbor koncnega zaporedja?