4. sklop: Algoritem Metropolis-Hastings

1 Primer: normalni model z znano varianco

Uporabili bomo algoritem Metropolis-Hastings za primer iz 3. sklopa, kjer so bili naši podatki število ur, ki so jih dijaki potrebovali za pripravo domače naloge.:

```
x <- c(2.11, 9.75, 13.88, 11.3, 8.93, 15.66, 16.38, 4.54, 8.86, 11.94,
  12.47, 11.11, 11.65, 14.53, 9.61, 7.38, 3.34, 9.06, 9.45, 5.98,
  7.44, 8.5, 1.55, 11.45, 9.73)
```

Privzemimo normalni model z znano varianco $\sigma^2 = 4$, torej $(X_i|\theta) \sim N(\theta, \sigma^2 = 4)$, medtem ko naj bo apriorna porazdelitev $\theta \sim N(\theta_0 = 6, \tau_0^2 = 9)$. Zanimala nas je aposteriorna porazdelitev $(\theta|X)$, ki vemo, da je porazdeljena normalno $N(\mu_1, \tau_1^2)$, kjer sta parametra enaka:

$$\mu_1 = \frac{\tau_0^2}{\frac{\sigma^2}{n} + \tau_0^2}\bar{X} + \frac{\frac{\sigma^2}{n}}{\frac{\sigma^2}{n} + \tau_0^2}\mu_0$$

$$\tau_1^2 = \frac{\frac{\sigma^2}{n} \cdot \tau_0^2}{\frac{\sigma^2}{n} + \tau_0^2}$$

Pravo aposteriorno porazdelitev torej poznamo.

V nalogi jo bomo aproksimirali s pomočjo algoritma Metropolis-Hastings.

2 Naloge za vaje, ki so hkrati domača naloga

Za primer iz 3. sklopa aproksimirajte aposteriorno porazdelitev s pomočjo algoritma Metropolis-Hastings, kjer sledite spodnjim korakom.

1. Sami v R-u sprogramirajte algoritem Metropolis-Hastings za naš primer. Izberite smiselno *predlagalno jedro* $q(\cdot|\theta^{(n-1)})$ (npr: $q(\cdot|\theta^{(n-1)}) \sim N(\theta^{(n-1)}, \sigma_q^2 = 0.1^2)$; lahko tudi izberete drugo porazdelitev). Ključno je, da algoritem sprogramirate sami, pri čemer splošnost kode in efektivnost implementacije nista pomembni. Opomba: $f(X|\theta)$ je verjetje za $X = (X_1, ..., X_n)$, kjer so X_i normalno porazdeljeni: $f(X|\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \cdot \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i - \theta)^2\right)$.
2. Preizkusite algoritem na našem primeru, kjer si sami izberite neko smiselno začetno vrednost $\theta^{(0)}$ in varianco *predlagalne gostote* (v zgornjem primeru smo jo označili σ_q^2). Opomba: zaradi numerične stabilnosti ob vsaki iteraciji izračunajte logaritem verjetnosti $\rho(\theta^{(n-1)}, y)$ in na podlagi tega logaritma se odločite, kakšen bo $\theta^{(n)}$. Rezultate predstavite na naslednji način:
   - Narišite celotno dobljeno zaporedje $\{\theta^{(0)}, \theta^{(1)}, ...\theta^{(S)}\}$ (naj bo S vsaj 30000 - lahko tudi vzamete več iteracij). Lahko uporabite funkcijo `plot(..., type='l')`.
   - Narišite le prvih 500 ali pa 5000 členov.
   - Narišite celotno zaporedje, kjer uporabite ustrezen *burn-in* parameter B.
   - Za tako izbrano zaporedje grafično predstavite aposteriorno porazdelitev in jo grafično primerjajte s pravo (teoretično) aposteriorno porazdelitvijo.
   - Ocenite parameter in 95% interval zaupanja za parameter iz izbranega zaporedja ter primerjajte z ocenami iz prave aposterirone porazdelitve.
3. Poženite vas algoritem pri neki nesmiselni zacetni vrednosti. Rezultate predstavite:
   - Narišite celotno dobljeno zaporedje $\{\theta^{(0)}, \theta^{(1)}, ...\theta^{(S)}\}$.
   - Narišite le prvih 500 ali pa 5000 členov.
   - Določite vrednost B, ki bi bila smiselna za vaš primer. Narišite celotno zaporedje, kjer uporabite ustrezen B.
4. Pri neki smiselni začetni vrednosti poženite algoritem pri nekaj razlicnih variancah za *predlagalno jedro*. Pri izboru pretiravajte v obe smeri (spomnite se, kakšni so po velikosti naši podatki), tako da boste grafično opazili razlike na prvih npr. 500 iteracijah. Rezultate predstavite:
   - Za vsak primer narisite prvih nekaj (nekje med 500 in 5000) členov in še celotno zaporedje.
   - Komentirajte razlike in zakaj do njih pride. Kaj in zakaj vas moti pri izbranih primerih?
   - Kakšen bi bil v splošnem (ne vezano na naš vzorec) vaš predlog glede izbora variance *predlagalnega jedra* oz. kakšen bi bil predlog za izbor končnega zaporedja?