

Bayes-päätely

Työterveyslaitos

8.-9.2.2018

Ville Hyvönen



4. MCMC-diagnostiikka

1. Trace plot
2. Scale reduction factor
3. Autokorrelaatio
4. Estimoitu otoskoko
5. Monte Carlo-keskivirhe

- ▶ MCMC - simulaatiossa simulaatio Markovin ketjusta tuottaa (autokorreloituneen, ei i.i.d.) otoksen posteriorijakaumasta, jos ketju on konvergoinut tasapainojakaumaansa.

- ▶ MCMC - simulaatiossa simulaatio Markovin ketjusta tuottaa (autokorreloituneen, ei i.i.d.) otoksen posteriorijakaumasta, jos ketju on konvergoinut tasapainojakaumaansa.
- ▶ Mistä tietää, onko ketju jo konvergoinut tasapainojakaumaansa, eli onko MCMC-otos edustava otos posteriorijakaumasta?

- ▶ MCMC - simulaatiossa simulaatio Markovin ketjusta tuottaa (autokorreloituneen, ei i.i.d.) otoksen posteriorijakaumasta, jos ketju on konvergoinut tasapainojakaumaansa.
- ▶ Mistä tietää, onko ketju jo konvergoinut tasapainojakaumaansa, eli onko MCMC-otos edustava otos posteriorijakaumasta?
 - ▶ 100%-varmasti ei mistään...

- ▶ MCMC - simulaatiossa simulaatio Markovin ketjusta tuottaa (autokorreloituneen, ei i.i.d.) otoksen posteriorijakaumasta, jos ketju on konvergoinut tasapainojakaumaansa.
- ▶ Mistä tietää, onko ketju jo konvergoinut tasapainojakaumaansa, eli onko MCMC-otos edustava otos posteriorijakaumasta?
 - ▶ 100%-varmasti ei mistään...
 - ▶ Ketjusta poistetaan alusta ns. burn-in period, esim. Stan poistaa automaattisesti ensimmäisen puolikkaan otoksesta.

- ▶ MCMC - simulaatiossa simulaatio Markovin ketjusta tuottaa (autokorreloituneen, ei i.i.d.) otoksen posteriorijakaumasta, jos ketju on konvergoinut tasapainojakaumaansa.
- ▶ Mistä tietää, onko ketju jo konvergoinut tasapainojakaumaansa, eli onko MCMC-otos edustava otos posteriorijakaumasta?
 - ▶ 100%-varmasti ei mistään...
 - ▶ Ketjusta poistetaan alusta ns. burn-in period, esim. Stan poistaa automaattisesti ensimmäisen puolikkaan otoksesta.
 - ▶ Konvergenssia voi tarkkailla ajamalla useamman ketjun eri alkuarvoista: jos kaikki ketjut liikkuvat samalla alueella, hyvin todennäköisesti ne ovat kaikki konvergoineet tasapainojakaumaansa (tietenkin on aina periaatteessa mahdollista, että *mikään* ketjuista ei ole konvergoinut).

- ▶ Stan ajaa oletusarvoisesti (argumentti `chains`) 4 Markovin ketjua rinnakkain.
 - ▶ Oletusarvoisesti (argumentti `iter`) jokaisessa ketjussa on 2000 iteraatiota, joten simuloidun otoksen koko on yhteensä 4000, koska puolet (tätä voi säätää argumentilla `warmup`) otoksesta hylätään burn-in / warmup-periodina.

- ▶ Stan ajaa oletusarvoisesti (`argumentti chains`) 4 Markovin ketjua rinnakkain.
 - ▶ Oletusarvoisesti (`argumentti iter`) jokaisessa ketjussa on 2000 iteraatiota, joten simuloidun otoksen koko on yhteensä 4000, koska puolet (tätä voi säätää argumentilla `warmup`) otoksesta hylätään burn-in / warmup-periodina.
- ▶ Ketjujen jälkeä (`trace`) parametriavaruudessa voi seurata trace plotista (`stan_trace`).
 - ▶ Jos ketjujen jäljet "sahaavat" päällekkäin, se on hyvä merkki.

- Ns. scale reduction factor (tai Gelman-Rubin statistic) \hat{R} vertaa ketjujen sisäistä varianssia niiden väliseen varianssiin.

- ▶ Ns. scale reduction factor (tai Gelman-Rubin statistic) \hat{R} vertaa ketjujen sisäistä varianssia niiden väliseen varianssiin.
- ▶ Jos \hat{R} on arvoltaan lähellä yhtä, niin ketjujen välillä ei ole juurikaan vaihtelua, eli ne ovat sekoittuneet hyvin.

- ▶ Ns. scale reduction factor (tai Gelman-Rubin statistic) \hat{R} vertaa ketjujen sisäistä varianssia niiden väliseen varianssiin.
- ▶ Jos \hat{R} on arvoltaan lähellä yhtä, niin ketjujen välillä ei ole juurikaan vaihtelua, eli ne ovat sekoittuneet hyvin.
- ▶ \hat{R} hat Stanin tulosteessa (jokaiselle parametrille erikseen).

- ▶ Ns. scale reduction factor (tai Gelman-Rubin statistic) \hat{R} vertaa ketjujen sisäistä varianssia niiden väliseen varianssiin.
- ▶ Jos \hat{R} on arvoltaan lähellä yhtä, niin ketjujen välillä ei ole juurikaan vaihtelua, eli ne ovat sekoittuneet hyvin.
- ▶ \hat{R} hat Stanin tulosteessa (jokaiselle parametrille erikseen).
- ▶ Jos \hat{R} hat ≤ 1.1 kaikille parametreille, niin trace plottia ei tarvitse yleensä tarkastella, vaan voidaan luottaa siihen että ketjut ovat konvergoineet.

- ▶ Ketjujen sekoittumista voi seurata myös estimoitujen posteriorijakauman tiheysfunktioiden kuvaajista, jotka piirretään jokaiselle 4:lle ketjulle erikseen.
 - ▶ Esimerkiksi parametrille `theta` komennolla `stan_dens(fit, 'theta', separate_chains = TRUE)`, missä `fit` sisältää `stan`-funktion palauttaman olion.

- Ketjujen sekoittumista voi seurata myös estimoitujen posteriorijakauman tiheysfunktioiden kuvaajista, jotka piirretään jokaiselle 4:lle ketjulle erikseen.
 - ▶ Esimerkiksi parametrille `theta` komennolla `stan_dens(fit, 'theta', separate_chains = TRUE)`, missä `fit` sisältää `stan`-funktion palauttaman olion.
 - ▶ Jos jokaisen ketjun perusteella estimoitu tiheysfunktio on lähes sama, se viittaa siihen, että kaikki ketjut ovat konvergoineet.

- ▶ Vaikka kaikki ketjut olisivat konvergoineet, se ei tarkoita, että simuloiminen olisi välttämättä *tehokasta*.

- ▶ Vaikka kaikki ketjut olisivat konvergoineet, se ei tarkoita, että simuloiminen olisi välttämättä *tehokasta*.
- ▶ Jos ketjun autokorrelaatio on suurta, eli sen peräkkäiset arvot ovat lähellä toisiaan, ketju tutkii parametriavaruutta vain hitaasti → tarvitaan paljon enemmän iteraatioita i.i.d. - otokseen verrattuna.

- ▶ Vaikka kaikki ketjut olisivat konvergoineet, se ei tarkoita, että simuloiminen olisi välttämättä *tehokasta*.
- ▶ Jos ketjun autokorrelaatio on suurta, eli sen peräkkäiset arvot ovat lähellä toisiaan, ketju tutkii parametriavaruutta vain hitaasti → tarvitaan paljon enemmän iteraatioita i.i.d. - otokseen verrattuna.
- ▶ Autokorrelaatiokuvasta (funktio `stan_ac`) näkee peräkkäisten arvojen väliset autokorrelaatiot (lägeille 0-30).
 - ▶ Mitä pienemällä lägillä autokorrelaatiot menevät (käytännössä) nollaan, sitä tehokkaammin ketju tutkii parametriavaruutta.

- ▶ Kuitenkaan autokorrelaatiokuvia tarvitsee harvoin tarkastella, sillä Stan tulostaa automaattisesti MCMC-otoksen **estimoidun otoskoon** (`n_eff` stan-funktion palauttaman olion `summary`-tulosteessa).

- ▶ Kuitenkaan autokorrelaatiokuvia tarvitsee harvoin tarkastella, sillä Stan tulostaa automaattisesti MCMC-otoksen **estimoidun otoskoon** (`n_eff` stan-funktion palauttaman olion `summary`-tulosteessa).
- ▶ Estimoitu otoskoko kuvaa, kuinka suurta i.i.d.-otosta autokorreloitunut MCMC-otos vastaa.

- ▶ Kuitenkaan autokorrelaatiokuvia tarvitsee harvoin tarkastella, sillä Stan tulostaa automaattisesti MCMC-otoksen **estimoidun otoskoon** (`n_eff` stan-funktion palauttaman olion `summary`-tulosteessa).
- ▶ Estimoitu otoskoko kuvaa, kuinka suurta i.i.d.-otosta autokorreloitunut MCMC-otos vastaa.
 - ▶ Esimerkkimallille MCMC-otoksen koko on $S = 20000$, mutta estimoitu otoskoko vain 6525.

- ▶ Kuitenkaan autokorrelaatiokuvia tarvitsee harvoin tarkastella, sillä Stan tulostaa automaattisesti MCMC-otoksen **estimoidun otoskoon** (`n_eff` stan-funktion palauttaman olion `summary`-tulosteessa).
- ▶ Estimoitu otoskoko kuvaa, kuinka suurta i.i.d.-otosta autokorreloitunut MCMC-otos vastaa.
 - ▶ Esimerkkimallille MCMC-otoksen koko on $S = 20000$, mutta estimoitu otoskoko vain 6525.
- ▶ Estimoitu otoskoko lasketaan jakamalla MCMC-otoksen koko S autokorrelaatioiden summalla:

$$\text{ESS} = \frac{S}{1 + 2 \sum_{i=1}^{\infty} \text{ACF}(i)},$$

missä $\text{ACF}(i)$ on keskimääräinen autokorrelaatio lägillä i .

- Jakauman häntien (esim. 95% uskottavuusvälit) tarkkaan estimointiin tulisi pyrkiä vähintään estimoituun otoskokoön 10000.

- ▶ Jakauman häntien (esim. 95% uskottavuusvälit) tarkkaan estimointiin tulisi pyrkiä vähintään estimoituun otoskokoön 10000.
- ▶ Jakauman keskellä olevien arvojen, kuten mediaanin (ja usein myös odotusarvon) tarkkaan estimointiin riittää pienempikin otos

- ▶ Monte Carlo-keskivirhe kuvaa Monte Carlo-estimaattien, esimerkiksi posteriorijakauman odotusarvon vaihtelu simuloitujen otosten välillä.

- ▶ Monte Carlo-keskivirhe kuvaa Monte Carlo-estimaattien, esimerkiksi posteriorijakauman odotusarvon vaihtelu simuloitujen otosten välillä.
- ▶ Huom. kuvaa sitä kuinka tarkasti simuloitujen otoksen perusteella approksimoitu tunnusluku, esimerkiksi posteriorijakauman odotusarvo kuvaa **tämän** nimenomaisen aineiston perusteella laskettua parametrin odotusarvoa, eikä sitä miten tarkasti posteriorijakauman odotusarvo kuvaa parametrin todellista arvoa (tämä riippuu tietenkin aineistosta)!

- ▶ Lasketaan jakamalla simuloidun otoksen keskihajonta sd estimoidun otoskoon neliöjuurella:

$$\text{MCSE} = \frac{\text{sd}}{\sqrt{\text{ESS}}}.$$

- ▶ Lasketaan jakamalla simuloidun otoksen keskihajonta sd estimoidun otoskoon neliöjuurella:

$$MCSE = \frac{sd}{\sqrt{ESS}}.$$

- ▶ Stan-tulosteessa `se_mean` (kunkin parametrin kohdalla).

- ▶ Lasketaan jakamalla simuloidun otoksen keskihajonta sd estimoidun otoskoon neliöjuurella:

$$MCSE = \frac{sd}{\sqrt{ESS}}.$$

- ▶ Stan-tulosteessa `se_mean` (kunkin parametrin kohdalla).
- ▶ Jos tarkoituksena on vain posteriorijakauman odotusarvon tarkan arvon selvittäminen, ei tarvitse välittää estimoidusta otoskoosta, vaan riittää tarkastella Monte Carlo-keskivirhettä.