

# Gibbsin otanta- ja Metropolis-Hastings algoritmit

Topias Karjalainen

3. toukokuuta 2020

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>2</b>
<b>2</b>	<b>Teoriaa</b>	<b>3</b>
2.1	Perusmääritelmiä . . . . .	3
2.2	Markovin ketjut . . . . .	4
<b>3</b>	<b>Markovin ketju Monte Carlo menetelmät</b>	<b>9</b>
3.1	Gibbsin otanta-algoritmi . . . . .	10
3.2	Metropolis–Hastings algoritmi . . . . .	12
3.2.1	Yhteys Gibsin otanta-algoritmiin . . . . .	15
3.2.2	Ehdotusjakauman valinnasta . . . . .	16
3.3	Yleisiä käytäntöjä MCMC-menetelmissä . . . . .	17
3.4	Konvergenssi ja Diagnostiikka . . . . .	17
3.4.1	Gelmanin $\hat{R}$ . . . . .	17
3.4.2	ESS . . . . .	18
<b>4</b>	<b>Laajempi esimerkki</b>	<b>20</b>
<b>5</b>	<b>Loppusanat</b>	<b>24</b>

# Luku 1

## Johdanto

Tilastotieteissä *frekventistinen* koulukunta oli pitkään vallitseva koulukunta. Viimeaikoina kuitenkin suosiotaan on kasvattanut Bayesilainen koulukunta. Aiemmin Bayesiläinen päättely ei päässyt leviämään, sillä toisin kuin frekventistinen koulukunta, Bayesiläisyys ei tarjonnut suurinpaan osaan kysymyksiä analyyttisiä ratkaisuja. Vasta tietokoneiden aikakautena *Markovin ketju Monte Carlo -menetelmät* (MCMC-menetelmät) ovat antaneet mahdollisuuden ratkaista *posteriori*-jakaumat monimutkaisemmilta malleilta.

Monte Carlo menetelmän kehittäjä 50-luvulla *Los Alamosissa* työskennelleet *Nicholas Metropolis*, *Stanislav Ulam* ja yleisnero *John von Neumann*. Yleinen määritelmä Monte Carlo menetelmälle on toistuva satunnainen arvojen arpominen. Yksinkertainen esimerkki Monte Carlo simuloinnista on esimerkiksi  $\pi$ :n arvon estimointi arpomalla sattumanvaraisesti pisteitä tasosta, ja laskemalla kuinka moni niistä on ympyrän säteen sisällä.

*Markovin ketjut* ovat *stokastisia prosesseja*, jotka on nimetty venäläisen matemaatikon *Andrey Markov*'n mukaan.

Tässä tutkielmassa tulen ensin antamaan lyhyen johdatuksen Markovin ketjuista ja selostan MCMC-menetelmien kannalta relevantin teorian. Tulen esittelemään lyhyesti kaksi algoritmia, joita käytetään MCMC-menetelmissä, *Gibbsin otanta-algoritmin* ja *Metropolis–Hastings algoritmin*. Esitän myös kaksi tärkeää diagnostiikkaa menetelmien tulosten arviointiin ja lopuksi vielä tarkastellaan hieman laajempaa käytännön esimerkkiä MCMC-algoritmista.

# Luku 2

## Teoriaa

### 2.1 Perusmääritelmiä

Määritellään ensiksi todennäköisyys.

**Määritelmä 2.1.**  $\sigma$ -algebra. Olkoot  $\Omega$  mielivaltainen epätyhjä joukko. Sigma-algebra perusjoukolla  $\Omega$  on sen osajoukkojen joukkoperhe  $\mathcal{F}$ , joka toteuttaa ehdot:

1.  $\emptyset \in \mathcal{F}$
2. jos  $A \in \mathcal{F}$ , niin  $A^c \in \mathcal{F}$
3. jos jos  $A_k \in \mathcal{F}$ , kaikilla  $k \in K$ , missä  $K$  on numeroituva joukko, niin  $\bigcup_{k \in K} A_k \in \mathcal{F}$

**Määritelmä 2.2.** Kuvaus  $\mathbf{P}$  liittää kuhunkin tapahtumaan  $A$  todennäköisyyden, joka on luku suljetulla välillä  $[0,1]$  ja sille pätee:

1.  $\mathbf{P}(\Omega) = 1$
2. Jos  $A$  on tapahtuma, niin sen komplementtitapahtuman  $A^c$  todennäköisyys on  $\mathbf{P}(A^c) = 1 - \mathbf{P}(A)$
3. Jos  $(A_k)_{k \in \mathbb{N}}$  ovat erillisiä tapahtumia, niin

$$\mathbf{P}\left(\bigcup_{k \in \mathbb{N}} A_k\right) = \sum_{k \in \mathbb{N}} \mathbf{P}(A_k)$$

**Määritelmä 2.3.** Olkoot  $\mathcal{A}$   $\sigma$ -algebra, ja olkoot  $X$  joukko. Pari  $(X, \mathcal{A})$  on mitallinen avaruus.

**Määritelmä 2.4.** Kolmikkoa  $(\Omega, \mathcal{F}, \mathbf{P})$  kutsutaan todennäköisyysavaruudeksi.

**Määritelmä 2.5.** Satunnaismuuttuja  $X$  on (lähes) mielivaltainen kuvaus  $X : \Omega \rightarrow S$ , jossa  $S$  on tilajoukko.

## 2.2 Markovin ketjut

Esitellään ensiksi joitain perus asioita Markovin Ketjuista, sillä ne eivät kuulu sellaisenaan opetussuunnitelmaan. Aloitetaan määrittelemällä stokastinen prosessi.

**Määritelmä 2.6.** Jono  $(X_n : n = 1, 2, 3, \dots)$  satunnaismuuttujia on diskreettiaikainen stokastinen prosessi.

**Merkintä 2.7.** Merkitään stokastista prosessia merkinnällä  $(X_n)$

Määritellään nyt siirtymäydin, eli jakauma, joka määrittelee tilojen välisten siirtymien todennäköisyydet ja esitetään sitten *Markovin ehto* diskreetille tila-avaruudelle määritelmässä 2.11, ja laajennetaan se jatkuvalle tila-avaruudelle määritelmässä 2.14. [7]

**Määritelmä 2.8.** Olkoot  $(S, \mathcal{S})$  ja  $(T, \mathcal{T})$  mitallisia avaruuksia. Siirtymäydin on funktio  $T : S \times \mathcal{T} \rightarrow [0, \infty]$ , jolle pätee

(i)  $\forall s \in S : A \rightarrow T(s, A)$  on todennäköisyyksimitta.

(ii)  $\forall A \in \mathcal{T} : s \rightarrow T(s, A)$  on mitallinen

Diskreetissä tapauksessa siirtymäyhtälöä kutsutaan *siirtymä matriisiksi*, joka on

$$(2.9) \quad p_{ij} = \mathbf{P}(X_n = i | X_{n-1} = j), \quad \forall i, j \in S$$

Jatkuvassa tapauksessayhtälö kuvaa sitä ehdollista todennäköisyyttä, että siirtymä tapahtuu, eli  $P(X \in A | x) = \int_A T(x, x') dx'$ . Tälle pätee, että

$$(2.10) \quad \int_S T(x, x') dx' = 1$$

**Määritelmä 2.11.** Stokastinen prosessi  $(X_n)$  on *Markovin ketjudiskreetissä tila-avaruudessa*, jos kaikilla alkuehdoilla  $m, n$  ja tiloilla  $i, j \in S$  on voimassa

$$(2.12) \quad \begin{aligned} & \mathbf{P}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) \\ &= \mathbf{P}(X_{n+1} = j | X_n = i) \end{aligned}$$

ja *siirtymätodennäköisyyksille* on voimassa

$$(2.13) \quad p_{ij} = \mathbf{P}(X_{n+1} = j | X_n = i) = \mathbf{P}(X_{m+1} = j | X_m = i)$$

Yhtälöä 2.12 kutsutaan *Markovin-ehdoksi* ja yhtälöä 2.13 taas kutsutaan *stationarisuusehdoksi*, mikä tarkoittaa, että siirtymätodennäköisyys tilojen  $i$  ja  $j$  välillä ei riipu ajasta  $m$  ja  $n$ , vaan pelkästään tiloista  $i$  ja  $j$ .

**Määritelmä 2.14.** Stokastinen prosessi  $(X_n)$  on *Markovin ketju* jatkuvassa tila-avaruudessa, jos kaikilla alkuhetkillä  $t$ ,  $X_t$ :n ehdolliselle jakaumalle pätee

$$(2.15) \quad P(X_t \in A | x_{t-1}, x_{t-2}, \dots, x_0) = P(X_t \in A | x_{t-1})$$

**Määritelmä 2.16.** Satunnaismuuttujan  $X_0$  jakaumaa kutsutaan *alkujakaumaksi*.

MCMC-menetelmien kannalta keskeinen ominaisuus Markovin ketjulle on sen tasapaino jakauma (eng. *invariant distribution*). Siihen perustuu koko idea menetelmän takana. Esitetään seuraavaksi tämä ominaisuus, sekä määritellään toinen ominaisuus eli *kääntävä Markovin ketju* esittämällä ns. *detailed balance*-yhtälö, jota tarvitaan Metropolis-hastings algoritmin tasapainojakauman olemassa olon osoittamiseen.

**Määritelmä 2.17.** Todennäköisyysjakauma  $\pi = (\pi)_{i \in S}$  on diskreetin tila-avaruuden Markovin ketjun  $(X_n)$  tasapainojakauma, jos

$$(2.18) \quad \sum_{i \in S} \pi_i p_{ij} = \pi_j, \forall j \in S$$

Yhtälö 2.18 voidaan kirjoittaa myös muotoon

$$(2.19) \quad \pi^T \mathbf{P} = \pi^T$$

Jakauma  $\pi$  on jatkuvan tila-avaruuden Markovin ketjun  $(X_n)$  tasapainojakauma jos

$$(2.20) \quad \pi(y) = \int_S \pi(x) T(x, y) dx$$

**Määritelmä 2.21.** Markovin ketju on *kääntävä*, jos löytyy sellainen TN-jakauma  $\lambda = (\lambda_i)_{i \in S}$ , että

$$(2.22) \quad \lambda_i p_{ij} = \lambda_j p_{ji}, \forall i, j \in S$$

**Määritelmä 2.23.** Markovin ketju jatkuvassa  $S$ :ssä on *kääntävä*, jos on olemassa

$$(2.24) \quad \pi(x) T(x, y) = \pi(y) T(y, x), \forall x, y \in S$$

Kääntyvällä Markovin ketjulla on sellainen mukava ominaisuus, että ketjun kääntyvyys on riittävä ehto tasapainojakauman olemassa ololle. Osoitamme tämän seuraavaksi.

**Lause 2.25.** Jos Markovin ketju on kääntävä, niin  $\lambda = \pi$  on sen tasapainojakauma.

*Todistus.*

$$\sum_{i \in S} \lambda_i p_{ij} = \sum_{i \in S} \lambda_j p_{ji} = \lambda_j \sum_{i \in S} p_{ij} = \lambda_j$$

□

**Lause 2.26.** Jos Markovin ketju  $(X_n)$  on kääntynvä ja tilajoukko  $S$  on jatkuva, niin  $\pi$  on sen tasapainojakauma.

*Todistus.* Yhtälön 2.10 mukaan  $\int_S T(y, x) dx = 1$ , joten

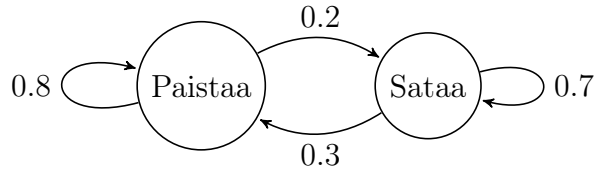
$$(2.27) \quad \int_S \pi(x) T(x, y) dx = \int_S \pi(y) T(y, x) dx = \pi(y) \int_S T(y, x) dx = \pi(y)$$

□

**Esimerkki 2.28.** Pohditaan lyhyttä esimerkkiä, jossa tilajoukko on  $S = \{”sataa”, ”paistaa”\}$ . Määritellään siirtymätodennäköisyydet siirtymämatriisilla

$$\mathbf{P}^{(1)} = \begin{pmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{pmatrix}$$

Tämä voidaan visualisoida kuvan 2.28 mukaisesti. Ketju on äärellinen, joten sillä on ta-



Kuva 2.1: Esimerkki 2.28

sapainojakauma. Yhtälö 2.19 implikoi, että jakauma  $\pi$  on siirtymämatriisin  $\mathbf{P}$  vasen ominaisvektori ( $\pi^T \mathbf{P} = \lambda \pi^T$ , jossa  $\lambda = 1$ ). Tämä voidaan ratkaista numeerisesti, ja ratkaisu on  $\pi^T = (0.4, 0.6)$ . Helposti nyt nähdään, että 2.19 pätee.

Seuraavaksi esitellään lyhyeksi *syklinen siirtymäydin*, ja osoitetaan tulos, jota tarvitsemme Gibbsin otanta-algoritmin tasapainojakauman todistuksen yhteydessä.

**Määritelmä 2.29.** Markovin ketjuissa voimme myös tietyn ehdoin yhdistää siirtymätiheyksiä. Tällöin puhutaan syklisistä siirtymätiheyksistä. Voidaan esimerkiksi määritellä siirtymätiheys

$$(2.30) \quad T_1 \dots T_d$$

**Lause 2.31.** Jos  $\pi$  on tasapainojakauma kaikille siirtymätiheyksille  $T_1, \dots, T_d$ , niin se on tasapainojakauma siirtymätiheydelle  $T = T_1 \dots T_d$ .

*Todistus.*

$$\pi T = \pi T_1 \dots T_d = \pi T_2 \dots T_d = \dots = \pi T_d = \pi$$

□

Käydään vielä nopeasti läpi muutama ominaisuus, joita tarvitaan kun haluamme osoittaa kappaleessa 3 algoritmimme toimivan.

**Määritelmä 2.32.** Markovin ketju  $(X_n)$ , siirtymäyhtymällä  $T(x, y)$  on *pelkistymätön* jos kaikilla  $A \in \mathcal{T}$ , joilla  $\mathbf{P}(A) > 0$ , on olemassa sellainen  $n$ , että  $T^n(x, A) > 0$  kaikilla  $x \in S$

Käytännössä pelkistymättömyys tarkoittaa siis sitä, että jokaisesta tila-avaruuden kolkasta on mahdollista päästä jokaiseen muuhun pisteeseen avaruutta, eli ketju ei voi jäädä jumiin johonkin alueelle.

**Määritelmä 2.33.** Markovin ketju  $(X_n)$  on *palautuva* jos

- (i) ketju  $(X_n)$  on pelkistymätön ja
- (ii) kaikilla  $A \in \mathcal{T}$ , joilla  $\mathbf{P}(A) > 0$ ,  $\mathbb{E}_x[\eta_A] = \infty$  kaikilla  $x \in A$

Missä  $\eta_A$  on käyntien määrä joukossa  $A$ .

Palautuvalla ketju tarkoittaa sitä, että ketju palaa alueelle, jossa se on jo käynyt. Palautumista vahvempi ominaisuus on *Harris-palautuvuus*.

**Määritelmä 2.34.** Joukko  $A$  on Harris palautuva, jos  $\mathbf{P}_x(\eta_A = \infty) = 1$  kaikilla  $x \in A$ . Markovin ketju  $(X_n)$  on Harris palautuva jos  $(X_n)$  on pelkistymätön, ja jokainen joukko  $A$ , jolla  $\mathbf{P}(A) > 0$ , on Harris palautuva.

**Määritelmä 2.35.** Markovin ketju  $(X_n)$  on *periodinen*, jos on olemassa erilliset osajoukot  $A_1, A_2, \dots, A_d \subset S$ ,  $d > 1$ , että

$$(2.36) \quad T(x, A_{i+1}) = 1, \quad \forall x \in A_i, \quad i = 1, \dots, d-1$$

ja

$$(2.37) \quad T(x, A_1) = 1, \quad x \in A_d$$

Jos ketju ei ole periodinen, se on *aperiodinen*.



Määritellään sitten vielä ergodisuus, joka on tärkeä ominaisuus Markovin ketjun tasapainojakauman olemassaolon kannalta.

**Määritelmä 2.38.** Markovin ketju  $(X_n)$  on ergodinen, jos se on pelkistymätön, aperiodinen ja Harris palautuva.

Ergodisuus on MCMC-menetelmien kannalta tärkeä ominaisuus, sillä se takaa Markovin ketjun  $(X_n)$  konvergoitumisen uniikkiin tasapainojakaumaansa mistä tahansa tilavaruuden  $S$  pisteestä. Tämän osoittaminen on melko hankalaa ja ylittää reilusti tämän tutkielman laajuuden, joten jätämme sen tekemättä.

# Luku 3

## Markovin ketju Monte Carlo menetelmät

Tässä luvussa aiomme esitellä MCMC-metodeja. Esittelemme alaluvussa 3.6 Gibbsin otanta-algoritmina tunnetun MCMC-menetelmän, ja sitten alaluvussa 3.2 esittelemme Metropolis–Hastings algoritmin. Osoitamme myös, että todellisuudessa alaluvun 3.6 algoritmi onkin todella vain erikoistapaus luvun 3.2 algoritmista. Pohditaan kuitenkin ensin menetelmän motivaatiota.

**Määritelmä 3.1.** MCMC-menetelmiksi kutsutaan jakaumaa  $p$  simuloiviin menetelmiin, jotka perustuvat siihen, että luodaan ergodinen Markovin ketju  $(X_n)$ , jolla on tasapainojakaumana jakauma  $p$ .

Ketjun ergodisuus takaa siis sen, että ketju konvergoituu jokaisesta tila-avaruuden pisteestä. Se takaa, että ketjun empiirinen keskiarvo

$$(3.2) \quad \mathfrak{J}_n = \frac{1}{n} \sum_{t=1}^n h(X_t)$$

konvergoituu odotusarvoon, eli

$$(3.3) \quad \lim_{n \rightarrow \infty} \mathfrak{J}_n \rightarrow \mathbb{E}_p[h(X_t)] = \int h(x)p(x)dx$$

Tällöin ketjun tiloja voidaan siis käsitellä i.i.d. otoksena tasapainojakaumasta.

Joillekin Markovin ketjuille (ja useimmille MCMC-menetelmien tuottamille Markovin ketjuille) pätee myös keskeinen raja-arvolause 3.4.

$$(3.4) \quad \sqrt{n}(\mathfrak{J}_n - \mathbb{E}_p[h(X_t)]) \rightarrow N(0, \sigma_h^2) \quad \text{kun } n \rightarrow \infty$$

jossa

$$(3.5) \quad \sigma_h^2 = \text{var}_p(h(X_0)) + 2 \sum_{t=1}^{\infty} \text{cov}_p(h(X_0), h(X_t))$$

Menetelmän ydin on siis rakentaa systemaattisella tavalla Markovin ketju, jonka tasapainojakaumana on haluamamme simuloitava jakauma. Tämän voisi kuvitella olevan kovin vaikeaa, mutta yllättävästi se onkin melko triviaalia.

### 3.1 Gibbsin otanta-algoritmi

*Gibbsin otanta-algoritmi* on tapa simuloida Bayesiläistä moniulotteista posteriorijakautta (eli ulottuvuuksia vähintään 2), kun suora otanta on hankalaa. Algoritmi on nimetty amerikkalaisen fyysikon, *Josiah Willard Gibbs*'n (1839-1903) mukaan, mutta sen todellinen kehittäjä on veljekset *Donald Geman* (1943-) ja *Stuart Geman* (1949-) vuonna 1984 artikkelissa *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*.

**Määritelmä 3.6.** Olkoot  $\theta$  parametrivektori, joka jaetaan  $d$ :hen osaan tai osavektoriin, eli  $\theta = (\theta_1, \theta_2, \dots, \theta_d)$ . Gibbsin otanta-algoritmi määritellään seuraavanlaisesti:

1. Valitaan satunnainen indeksi  $j$  jolle  $1 \leq j \leq d$
2. Arvotaan uusi tila jokaiselle osavektorille  $\theta_j$  ehdollistamalla se jokaiselle muulle parametrille, eli vedetään arvot  $\theta = (\theta_1, \theta_2, \dots, \theta_d)$  jakaumista

$$(3.7) \quad p(\theta_j | \theta_{-j}^{(n-1)}, y)$$

missä  $\theta_{-j}^{(n-1)}$  on kaikki muut  $\theta$ :n komponentit paitsi  $j$ s komponentti, näiden tämänhetkisillä arvoilla eli

$$\theta_{-j}^{(n-1)} = (\theta_1^{(n-1)}, \dots, \theta_{j-1}^{(n-1)}, \theta_{j+1}^{(n-1)}, \dots, \theta_d^{(n-1)})$$

Yleensä ylläolevassa määritelmässä kohdassa 1. ei arvota uutta järjestystä, vaan järjestys päätetään alussa, ja sitä pidetään koko algoritmin ajan samana. Tällöin kyseessä on ns. *systemic scan Gibbs sampler*.

**Lause 3.8.** Määritelmän 3.6 mukaisen algoritmin tuottama Markovin ketju on aperiodinen, pelkistymätön ja Harris palautuva ja sillä on sitten uniikki tasapainojakauma  $p(\theta)$ . [5]

*Todistus.* Voitaisiin osoittaa, että määritelmän 3.6 algoritmin tuottama Markovin ketju on ergodinen, ja täten konvergoituu tasapainojakaumaan. Ohitetaan se kuitenkin nyt tässä, sillä todistus on melko tekninen.

Olkoot  $\theta^{(n-1)}$  alkuperäisen tila, ja  $\theta_j^{(n)}$  uusi  $j$ :nennen parametrin tila. Nyt  $\theta^{(n-1)}$ :n ja  $\theta_j^{(n)}$ :n yhteis tiheys on

$$(3.9) \quad p(\theta^{(n-1)}) p_j(\theta_j^{(n)} | \theta_{-j}^{(n-1)})$$

Nyt voidaan integroida

$$\begin{aligned}
(3.10) \quad \int p(\theta^{(n-1)})p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)})d\theta_j^{(n-1)} &= \int p(\theta_j^{(n-1)}|\theta_{-j}^{(n-1)})p(\theta_{-j}^{(n-1)})p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)})d\theta_j^{(n-1)} \\
&= p(\theta_{-j}^{(n-1)})p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)}) \int p(\theta_j^{(n-1)}|\theta_{-j}^{(n-1)})d\theta_j^{(n-1)} \\
&= p(\theta_{-j}^{(n-1)})p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)}) \\
&= p(\theta_j^{(n)}, \theta_{-j}^{(n-1)})
\end{aligned}$$

Eli Gibbs otanta-algoritmin päivitys ei muuta jakaumaa. Nyt voidaan soveltaa lausetta 2.31, jolloin voidaan todeta, että koska  $p$  tasapainojakauma jokaiselle  $p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)})$ , niin se on tasapainojakauma niiden yhteis siirtymätiheydelle

$$(3.11) \quad T = \prod_{j=1}^d p_j(\theta_j^{(n)}|\theta_{-j}^{(n-1)})$$

□

Ohitetaan Gibbsin otanta-algoritmin kohdalla toistaiseksi esimerkit, ja palataan siihen kappaleessa 4, jossa tarkastelemme laajempaa esimerkkiä lineaarisen regression parissa. Toteutamme tämän Gibbsin otanta-algoritmina.

Gibbsin otanta-algoritmin vaatimus ehdollisista jakaumista voi vaikuttaa jokseenkin rajoittavalta, sillä joskus voidaan tutkia mallia, jolle on vaikea laskea ehdolliset jakaumat. Tällöin voidaan kuitenkin pienin muunnoksia hyödyntää Gibbsin otanta-algoritmia hyödyntämällä seuraavaa määritelmää 3.12.

**Määritelmä 3.12.** Olkoot  $p$  tiheysfunktio. Tiheysfunktio  $\lambda$ , joka täyttää ehdon

$$(3.13) \quad \int_Z \lambda(x, z)dz = p(x)$$

kutsutaan  $p$ :n täydellistymäksi (*completion*).

Tiheys  $\lambda$  valitaan siten, että sille voi ratkaista ehdolliset jakaumat. Nyt Gibbsin otanta-algoritmi voidaan toteuttaa käyttämällä jakaumaa  $\lambda$  ja sen ehdollisia jakaumia. Lauseen 3.8 tulos pätee myös tässä tapauksessa. Tällöin malliin tulee toki uusi parametri  $z$ , joka on täysin turha mallin kannalta, mutta avustaa simuloinnissa.

## 3.2 Metropolis–Hastings algoritmi

*Metropolis–Hastings* algoritmi on kehittäjööidensä *Nicholas Metropolisksen* (1915–1999) ja *Wilfred Keith Hastingsin* (1930–2016) mukaan nimetty MCMC-menetelmä, jolla voidaan simuloida Bayesiläisessä analyysissä käytettäviä posteriori jakaumia myös silloin kun tiheys on mahdotonta määrittää analyttisesti.

Algoritmin pohjan kehitti *Stanislav Ulam* ja *Metropolis* työskennellessään *Los Alamosissa* ja myöhemmin *Metropolis* kehitti nykyään *Metropolis-algoritmina* tunnettua algoritmia ja esittelivät sen artikkelissa *Equation of state calculations by fast computing machines* [6]. Tämä versio algoritmista vaati, että pian esiteltävä *ehdotusjakauma* on symmetrinen. Myöhemmin *Hastings* laajensi algoritmin koskemaan myös epäsymmetrisiä ehdotusjakaumia artikkelissa *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*

**Merkintä 3.14.** TN-jakauma  $J_n(\cdot|\cdot)$  on niin sanottu *ehdotusjakauma* (*proposal distribution*, *jumping distribution*), josta *MH-algoritmissa* arvotaan ehdotus tila.

**Määritelmä 3.15.** Metropolis–Hastings algoritmi on seuraavanlainen

1. Valitaan aloitus tila  $\theta_0$  ja asetetaan  $n = 0$
2. Generoidaan kandidaatti tila  $\theta'$  satunnaisesti jakaumasta  $J_n(\theta'|\theta_{n-1})$
3. Lasketaan suhde

$$r = \frac{p(\theta'|y)/J_n(\theta'|\theta_{n-1})}{p(\theta_{n-1}|y)/J_n(\theta_{n-1}|\theta')}$$

4. Asetetaan

$$\theta_t = \begin{cases} \theta', & \text{todennäköisyydellä } \min(r, 1) \\ \theta_{t-1}, & \text{muuten} \end{cases}$$

Jossa  $J_t(\theta'|\theta^{t-1})$  on ns. ehdotusjakauma (eng. proposal distribution).

**Lause 3.16.** Määritelmän 3.15 algoritmi tuottaa Markovin ketjun jolla on uniikki tasapainojakauma  $p(\theta)$

*Todistus.* Todistus nojautuu Markovin ketjun kääntyvyysominaisuuteen (2.21 ja 2.23), eli

$$(3.17) \quad T(\theta^{(n)}|\theta^{(n-1)})p(\theta^{(n-1)}) = T(\theta^{(n-1)}|\theta^{(n)})p(\theta^{(n)})$$

joka on siis riittävä ehto tasapainojakauman olemassaololle. Mietitään kahta tapausta: (1)  $\theta^{(n)} \neq \theta^{(n-1)}$  ja (2)  $\theta^{(n)} = \theta^{(n-1)}$ . Tapauksen (2) siirtymä voi tapahtua kahdella tavalla. Josko kohdassa 4. ehdotus  $\theta'$  hylätään, tai se hyväksytään, mutta osutaan sattumanvaraisesti takaisin samaan kohtaan. Kuitenkin selvästi nähdään, että ehto 3.17 pätee tilanteessa (2).

Tilanteessa (1) siirtymätodennäköisyys pisteestä  $\theta^{(n-1)}$  pisteeseen  $\theta^{(n)}$  on

$$(3.18) \quad T(\theta^{(n)}|\theta^{(n-1)}) = J_n(\theta^{(n)}|\theta^{(n-1)}) \min \left( \frac{p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)})}{p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)})}, 1 \right)$$

Jota voidaan muokata helposti

$$(3.19) \quad \begin{aligned} T(\theta^{(n)}|\theta^{(n-1)}) &= J_n(\theta^{(n)}|\theta^{(n-1)}) \min \left( \frac{p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)})}{p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)})}, 1 \right) \\ &= \frac{1}{p(\theta^{(n-1)})} \min \left( p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)}), p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)}) \right) \end{aligned}$$

Nähdään kuitenkin, että yhtälön 3.19 alempi yhtäläisyys on symmetrinen eli

$$(3.20) \quad T(\theta^{(n-1)}|\theta^{(n)}) = \frac{1}{p(\theta^{(n)})} \min \left( p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)}), p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)}) \right)$$

joten kerrotaan 3.19 termillä  $p(\theta^{(n-1)})$  ja hyödynnetään 3.20 ominaisuutta

$$\begin{aligned} T(\theta^{(n)}|\theta^{(n-1)})p(\theta^{(n-1)}) &= \frac{1}{p(\theta^{(n-1)})} \min \left( p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)}), p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)}) \right) p(\theta^{(n-1)}) \\ &= \frac{1}{p(\theta^{(n)})} \min \left( p(\theta^{(n-1)})J_n(\theta^{(n)}|\theta^{(n-1)}), p(\theta^{(n)})J_n(\theta^{(n-1)}|\theta^{(n)}) \right) p(\theta^{(n)}) \\ &= T(\theta^{(n-1)}|\theta^{(n)})p(\theta^{(n)}) \end{aligned}$$

Eli myös tapauksessa (1) yhtälö 3.17 pätee. □

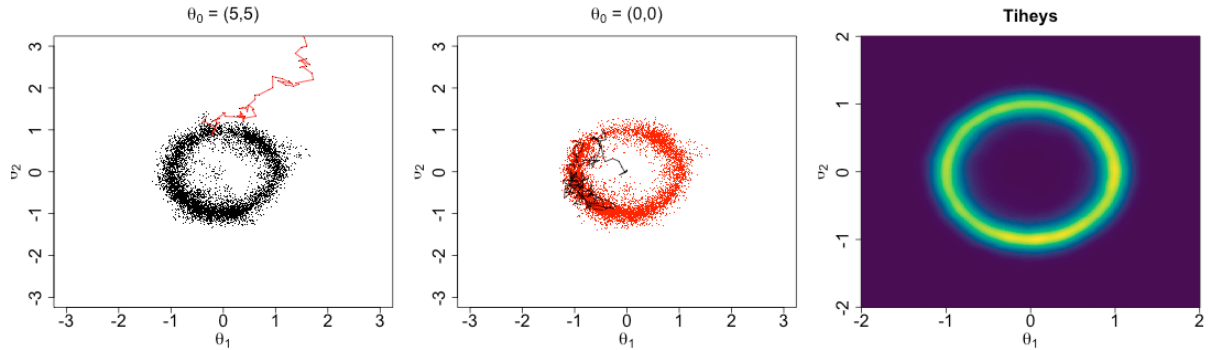
**Esimerkki 3.21.** Ajatellaan kuvitteellista tapausta, jossa meillä jatkuva kaksiulotteinen todennäköisyysjakauma, jonka tiheysfunktio on

$$(3.22) \quad p(\theta) \propto \exp(-5|\theta_1^2 + \theta_2^2 - 1|)$$

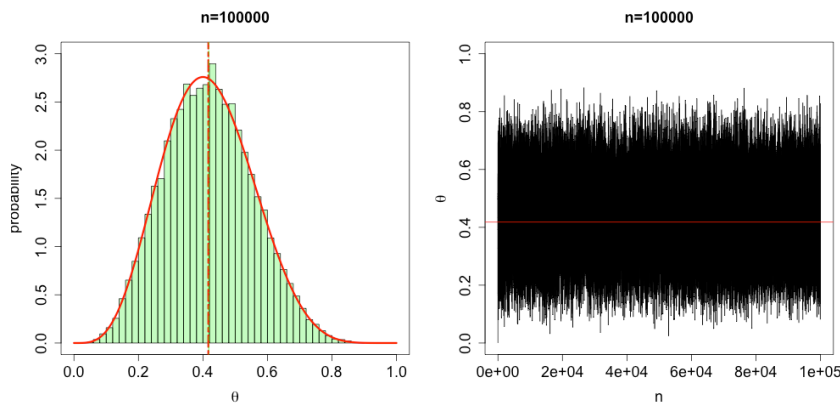
joka muodostaa regasmaisen 2-ulotteisen jakauman. Valitaan ehdotusjakaumaksi  $J_n(\theta_n|\theta_{n-1})$  2d-multinormaalijakauma

$$(3.23) \quad J_n(\theta_n|\theta_{n-1}) \sim N(\theta_{n-1}, \sigma^2 I_2)$$

jossa  $I_2$  on 2x2 yksikkömatriisi ja olkoot  $\sigma^2 = 0.01$ . Nyt Metropolis Hastings algoritmin avulla voidaan simuloida jakaumaa  $p(\theta)$  algoritmilla 3.15. Simuloidaan kaksi Markovin ketjua asettamalla aloitustiloiksi  $(0, 0)$  ja  $(5, 5)$ , kummastakin 10 000 tilaa. Simuloimme myös 200 000 pistettä aloitusarvolla  $(0, 0)$ , joista luodaan tiheysestimaatti. Tulokset löytyy kuvasta 4.2. Kahdessa ekassa kuvassa viiva on ensimmäisen 250 pisteen polku. Selvästi nähdään, että aloituspisteellä ei ole väliä. Markovin ketjun tasapainojakauma on sama huolimatta aloituspisteestä.



Kuva 3.1: Vasemmalla:  $(5,5)$ . Keskellä:  $(0,0)$  Oikealla: tiheysestimaatti (huomaa eri skaalaa).



Kuva 3.2: Esimerkin 3.24 tulokset

**Esimerkki 3.24.** Otetaan toisena esimerkkinä klassinen tapaus, jossa oletetaan, että havainnot ovat jauakautuneet *Bernoulli-jakauman* mukaan,  $y_i \sim \text{Bernoulli}(\theta)$ , ja että prior on tasajakauma. Tällöin tiedetään, että analyttinen posteriori on

$$p(\theta|y_i) = \text{Beta}\left(\sum_{i=1}^n y_i + 1, n - \sum_{i=1}^n y_i + 1\right)$$

Valitaan hieman eksoottinen ehdotusjakauma esimerkin vuoksi:

$$(3.25) \quad J_n(\theta_n|\theta_{n-1}) \sim \begin{cases} \text{Unif}(\theta_{n-1}, 1) & \text{kun } \theta_{n-1} < 0.5 \\ \text{Unif}(0, \theta_{n-1}) & \text{kun } \theta_{n-1} \geq 0.5 \end{cases}$$

Toisin kuin esimerkissä 3.21, nyt ehdotusjakauma ei olekkaan symmetrinen.

Oletetaan että, meillä on havainnot  $(1, 1, 1, 0, 0, 1, 0, 0, 0, 0)$  ja tarkastellaan sekä analyttistä että MH-algoritmin tuottamaa jakaumaa ja niiden eroja.

Kuvasta 3.2 nähdään esimerkin tulokset. Huomataan, että vaikka ehdotusjakauma on melko kummallinen, niin kuitenkin tarpeeksi monella iteraatiolla saavutetaan tasapainojakauma. Huomaa, että vasemmassa kuvaajassa on vihreällä simulaatio keskiarvo, ja punaisella analyttinen keskiarvo, mutta nämä arvot ovat niin lähellä toisiaan, että viivat ovat päällekkäin.

### 3.2.1 Yhteys Gibbsin otanta-algoritmiin

Kuten jo totesimme aiemmin, Gibbsin otanta-algoritmin voi esittää Metropolis-Hastings algoritmin erityis tilanteena, jossa hyväksymis todennäköisyys on 1.

**Lause 3.26.** *Gibbsin otanta-algoritmi on erityis tapaus Metropolis-hastings algoritmista.*

*Todistus.* Määritellään MH-algoritmin iteraatio  $n$  niin, että se sisältää  $d$  askelta, ja jossa iteraation  $n$  askel  $j$  kuvaa parametriosavektorin  $\theta_j$  päivitystä ehdollistettuna kaikilla muilla  $\theta$ :n elementeillä. Tällöin ehdotusjakauma  $J_{j,n}$  iteraation  $n$  askeleessa  $j$ , voidaan määrittää siten, että se ehdottaa vain tiloja, jotka muuttaa vain parametria  $\theta_j$ . Eli:

$$(3.27) \quad J_{j,n}^{Gibbs}(\theta^*|\theta^{(n-1)}) = \begin{cases} p(\theta_j^*|\theta_{-j}^{(n-1)}) & \text{kun } \theta_{-j}^* = \theta_{-j}^{(n-1)} \\ 0 & \text{muulloin} \end{cases}$$

Nyt voidaan kirjoittaa suhde  $r$  iteraation  $n$   $j$ :nen askeleen kohdalla muotoon

$$(3.28) \quad \begin{aligned} r &= \frac{p(\theta^*)/J_{j,n}^{Gibbs}(\theta^*|\theta^{(n-1)})}{p(\theta^{(n-1)})/J_{j,n}^{Gibbs}(\theta^{(n-1)}|\theta^*)} \\ &= \frac{p(\theta^*)/p(\theta_j^*|\theta_{-j}^{(n-1)})}{p(\theta^{(n-1)})/p(\theta_j^{(n-1)}|\theta_{-j}^{(n-1)})} \\ &= \frac{p(\theta_{-j}^{(n-1)})}{p(\theta_{-j}^{(n-1)})} \\ &= 1 \end{aligned}$$

Eli määrittelemällä MH-algoritmile siirtymäjakauma sillä tavalla, että se vastaa Gibbsin otanta-algoritmia, siirtymän hyväksymistodennäköisyydeksi tulee 1.  $\square$



### 3.2.2 Ehdotusjakauman valinnasta

Kummassakin kappaleen 3.2 esimerkissä valitsimme ehdotusjakauman melko satunnaisesti. Varsinkin esimerkissä 3.24 se on erittäin epätavallinen, mistä syystä hyvän approksimaation saavuttaminen vie todella monta iteraatiota. Yleensä jos haluamme oikeasti tehokkaasti ja ekonomisesti simuloida jakaumia esitetyllä algoritmilla, haluamme valita ehdotusjakauman jollakin järkevällä, systemaattisella tavalla, joka minimoisi tarvittavien iteraatioiden määrän.

Yleisesti ottaen hyvällä ehdotusjakaumalla on muutama ominaisuus[1]

1. Kaikilla  $\theta$ :n arvoilla on helppo arpoa arvo  $J(\theta'|\theta)$
2. Suhde  $r$  on helppo laskea
3. Siirtymät ovat tarpeeksi pitkiä. Muuten Markovin Ketju etenee liian hitaasti ja hyvän estimaatin saaminen kestää liian pitkään.
4. Siirtymiä ei hylätä liian usein. Muuten Markovin Ketju ei etene vaan seisoo paikallaan.

Lisäksi simulointia voidaan nopeuttaa mm. käyttämällä adaptiivista ehdotusjakaumaa, eli toisinsanoen ehdotusjakaumaa muunnellaan riippuen ketjun liikkeistä.

### 3.3 Yleisiä käytäntöjä MCMC-menetelmissä

Usein on tapana simuloida useampi kuin yksi ketju kustakin tapauksesta. Tällöin asetetaan näiden ketjujen aloitustilat eriäviksi, jotta saadaan vähennettyä aloitustilan vaikutusta lopputulokseen.

Toinen yleinen käytäntö on jättää ketjun alkupäästä jonkin verran tiloja huomiotta, sillä ketjun alkupäässä ei se ei välttämättä ole vielä saavuttanut tasapainojakaumaa. Tätä kutsutaan *Burn-in periodiksi*.

Joskus taas on hyvä tapa pudottaa joka  $n$ :s tila ketjusta. Tilojen tiputtaminen ei vaikuta tasapainojakaumaan, kunhan ketju vain on saavuttanut sen. Tilojen tiputtamista on hyötyä jos mallissa on paljon parametreja, jolloin tietokoneessa voi tulla ongelmia tilan kanssa.

### 3.4 Konvergenssi ja Diagnostiikka

Kappaleissa 3.6 ja 3.2 esitetyjen algoritmien kohdalla voi herätä kysymys, että mikä on riittävä määrä iteraatioita, jotta Markovin ketju on saavuttanut tasapainojakaumansa ja otanta on riittävän hyvä aproksimaatio posteriorijaukaumasta. Esimerkiksi kun katsotaan kuvan 4.2 vasemman puolen kuvan punaista polkua, niin voimme sanoa, että se ei ole vielä saavuttanut tasapainojakaumaa sillä tunnemme melko hyvin halutun jakauman, mutta yleensä emme välttämättä osaa sanoa tätä suoraan.

Toinen yleinen ongelma, joka kaipa pohdintaa on se, että ketjujen sisällä on korrelaatiota, mikä vaikeuttaa päättelyä simuloinnin tuloksista. Otannat eivät siis ole välttämättä täysin riippumattomia.

#### 3.4.1 Gelmanin $\hat{R}$

Yksi yleisimmistä estimaattoreista, joita käytetään MCMC-metodeissa Markovin ketjujen konvergenssin arvioimiseen on *Andrew Gelmanin*  $\hat{R}$  [1]. Se mittaa ketjujen sisäistä sekoittumista (eng. *mixing*) ja erillisten ketjujen välistä sekoittumista.

**Määritelmä 3.29.** *Gelmanin*  $\hat{R}$  [2][3] lasketaan jakamalla ensin jokin määrä simuloituja ketjua erillisillä aloituspisteillä keskeltä kahtia. Olkoon nyt  $m$  ketjujen määrä jaon jälkeen ja  $n$  ketjujen pituus. Olkoot  $\psi_{ij}(i = 1, \dots, n; j = 1, \dots, m)$  tila  $i$  ketjussa  $j$ . Nyt merkitään

$$(3.30a) \quad B = \frac{n}{m-1} \sum_{j=1}^m (\bar{\psi}_{\cdot j} - \bar{\psi}_{\cdot \cdot})^2, \quad \text{jossa}$$

$$(3.30b) \quad \bar{\psi}_{.j} = \frac{1}{n} \sum_{i=1}^n \psi_{ij}$$

$$(3.30c) \quad \bar{\psi}_{..} = \frac{1}{m} \sum_{j=1}^m \bar{\psi}_{.j}$$

Ja merkataan

$$(3.31) \quad W = \frac{1}{m} \sum_{j=1}^m s_j^2, \text{ jossa } s_j^2 = \frac{1}{n-1} \sum_{i=1}^n (\psi_{ij} - \bar{\psi}_{.j})^2$$

Jossa siis  $B$  on ketjujen välinen varianssi (*between sequence*) ja  $W$  on ketjujen sisäinen varianssi (*within sequence*). Näiden painotettuna keskiarvona saadaan estimaattori  $\hat{\psi}$ :n marginaaliselle posteriori varianssille

$$(3.32) \quad \hat{\sigma}^+(\psi|y) = \frac{n-1}{n}W + \frac{1}{n}B$$

Nyt  $\hat{R}$  voidaan määritellä

$$(3.33) \quad \hat{R} = \sqrt{\frac{\hat{\sigma}^+(\psi|y)}{W}}$$

Gelmanin  $\hat{R}$  mittaa sitä miten paljon kullakin hetkellä  $\psi$ :n jakauma voisi supistua, jos simulaatioiden annettaisiin jatkua loputtomasti. Eli siis kun  $\hat{R} \approx 1$ , niin voidaan sanoa, että Markovin ketju on todennäköisesti saavuttanut tasapainojakaumansa, sillä tällöin eri pisteitä aloitettujen ketjujen välinen varianssi on vakaantunut.

Tällä diagnostiikalla on kuitenkin ongelmansa ja rajoitteensa. Se saattaa virheellisesti diagnosoida konvergoituneen ketjun esimerkiksi jos ketjulla on pitkät hännät. Sen takia käytetäänkin mielummin muita diagnostiikka keinoja. Esimerkiksi *Stan* käyttää korjattua  $\hat{R}$  diagnostiikkaa [8].

### 3.4.2 ESS

Toinen hyödyllinen diagnostiikka MCMC-menetelmissä on niin sanottu *effective sample size (ESS)*, joka mittaa sitä kuinka monta efektiivisesti riippumatonta otosta ketjussa on. Tämä tarkoittaa siis sitä, että se mittaa kuinka paljon ketjun autokorrelaatio vaikuttaa keskivirheeseen verrattuna täysin riippumattomiin otoksiin.

**Määritelmä 3.34.** ESS lasketaan kaavalla

$$(3.35) \quad \hat{n}_{eff} = \frac{mn}{1 + 2 \sum_{t=1}^T \hat{\rho}_t}$$

jossa

$$(3.36a) \quad \hat{\rho}_t = 1 - \frac{V_t}{2\hat{\sigma}^+}$$

$$(3.36b) \quad V_t = \frac{1}{m(n-t)} \sum_{j=1}^m \sum_{i=t+1}^n (\psi_{ij} - \psi_{i-t,j})^2$$

$\hat{\sigma}^+$  saadaan kaavasta 3.32. Termissä  $\sum_{t=1}^T \hat{\rho}_t$  autokorrelaatioita summataan, kunnes kahden peräkkäisen kovariaatin summa on negatiivinen [4].

# Luku 4

## Laajempi esimerkki

Tarkastellaan vielä lopuksi laajempaa esimerkkiä. Tarkastellaan normaalia lineaarista regressiomallia. Käytetään R:stä löytyvää `airquality` dataa, jossa on dataa New Yorkin ilmanlaadusta. Regressoidaan datasetin muuttuja `Ozone` muuttujille `Solar.R` ja `Wind`. Sovitetaan tämä malli Gibbsin otanta-algoritmin avulla. Sitä varten meidän on ensiksi määriteltävä marginaaliset ehdolliset jakaumat parametreille.

Määritellään ensiksi malli. Merkitään  $\tau = 1/\sigma^2$ ,  $\theta = (\beta_0, \beta_1, \beta_2, \tau)$

$$\begin{aligned} y_i | \beta_0, \beta_1, \beta_2, \tau &\sim N(\beta_0 + \beta_1 x_i + \beta_2 x_i, 1/\tau) \\ \beta_0 | \mu_0, \tau_0 &\sim N(\mu_0, 1/\tau_0) \\ \beta_1 | \mu_1, \tau_1 &\sim N(\mu_1, 1/\tau_1) \\ \beta_2 | \mu_2, \tau_2 &\sim N(\mu_2, 1/\tau_2) \\ \tau | \alpha, \gamma &\sim \text{Gamma}(\alpha, \gamma) \end{aligned} \tag{4.1}$$

Muodostetaan uskottavuusfunktio

$$L(y|\theta) = \prod_{i=1}^N N(\beta_0 + \beta_1 x_i + \beta_2 x_i, 1/\tau) \tag{4.2}$$

Ja posterioiri on siten

$$p(\theta|y) \propto p(\beta_0)p(\beta_1)p(\beta_2)p(\tau) \prod_{i=1}^N N(\beta_0 + \beta_1 x_i + \beta_2 x_i, 1/\tau) \tag{4.3}$$

Tästä saadaan sitten johdettua marginaaliset jakaumat otanta-algoritmia varten

$$\begin{aligned}
(4.4) \quad & \beta_0 | \beta_1, \beta_2, \tau_0, \tau, \mu_0, x, y \sim N \left( \frac{\tau_0 \mu_0 + \tau \sum_{i=1}^N (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})}{\tau_0 + \tau N}, \frac{1}{\tau_0 + \tau N} \right) \\
& \beta_1 | \beta_0, \beta_2, \tau_1, \tau, \mu_1, x, y \sim N \left( \frac{\tau_1 \mu_1 + \tau \sum_{i=1}^N (y_i - \beta_0 - \beta_2 x_{i2}) x_{i1}}{\tau_1 + \tau \sum_{i=1}^N x_{i1}^2}, \frac{1}{\tau_1 + \tau \sum_{i=1}^N x_{i1}^2} \right) \\
& \beta_2 | \beta_0, \beta_1, \tau_2, \tau, \mu_2, x, y \sim N \left( \frac{\tau_2 \mu_2 + \tau \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_{i1}) x_{i2}}{\tau_2 + \tau \sum_{i=1}^N x_{i2}^2}, \frac{1}{\tau_2 + \tau \sum_{i=1}^N x_{i2}^2} \right) \\
& \tau | \beta_0, \beta_1, \beta_2, \alpha, \gamma, x, y \sim \text{Gamma} \left( \alpha + \frac{N}{2}, \gamma + \frac{N}{2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2})^2 \right)
\end{aligned}$$

Asetetaan hyperparametrit  $\mu_0 = 80, \mu_1 = 0, \mu_2 = -5, \tau_0 = 1/50, \tau_1 = 1/50, \tau_2 = 1/50, \alpha = 5, \gamma = 0.01$ . Parametrit on valittu sen mukaan, että ne asettavat suurimman tiheyden lähelle arvioitua sijaintia ja toisaalta eivät ole kovin informatiivisia vaan leveitä. Simuloidaan nyt tästä Gibbsin otanta-algoritmillä yhtälöitä 4.4 käyttäen 8 ketjua, kunkin pituus 2 000. Burnin-periodi olkoot 2000. Yhteensä siis meillä on 16 000 otosta.

$\theta$	Mean	SD	2.5%	25%	50%	75%	97.5%
$\beta_0$	78.89544	5.61842	67.97969	75.14744	78.89385	82.64874	89.86483
$\beta_1$	0.09675	0.02263	0.05196	0.08130	0.09688	0.11218	0.14106
$\beta_2$	-5.48880	0.51291	-6.48407	-5.83627	-5.49158	-5.13095	-4.48973
$\tau$	0.00177	0.00023	0.00136	0.00161	0.00176	0.00192	0.00226

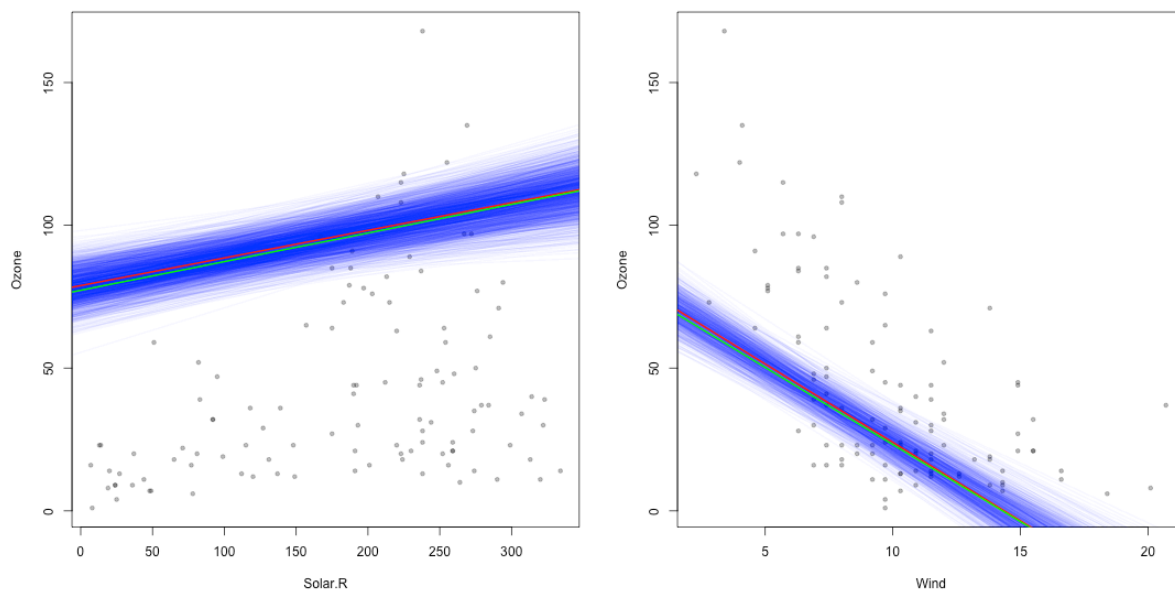
Taulukko 4.1: Tulokset regressiosta

$\theta$	$\hat{R}$	$\hat{n}_{eff}$
$\beta_0$	1.00	778
$\beta_1$	1.00	1177
$\beta_2$	1.01	902
$\tau$	1.00	7656

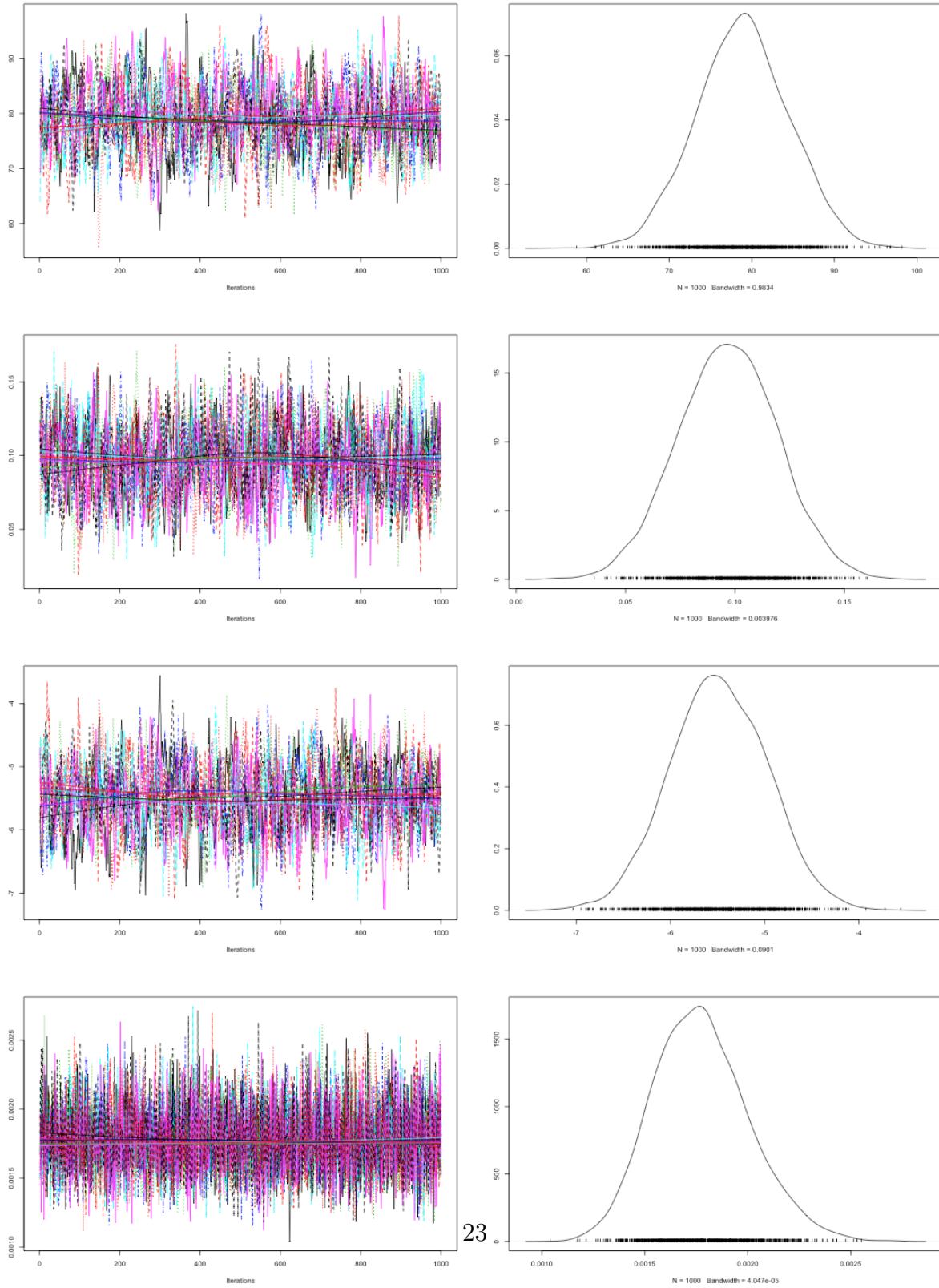
Taulukko 4.2: Diagnostiikka

Taulukoissa 4.1 ja 4.2 nähdään tulokset ja MCMC-diagnostiikat, joista puhuttiin aiemmin. Nähdään, että ketjut näyttäisivät konvergoituneen, mikä ei sinänsä ihmetytä lainkaan, sillä simulaatio määrämme on valtava. Yhteensä burn-in periodin jälkeen otoksia on 8000.

Mielenkiintoista on nähdä, että kuvassa 4.2 ketju parametrille  $1/\tau$  on aivan ernäköinen kuin muille parametreille. Otoksia on yhtä paljon kaikissa, mutta  $1/\tau$ :n ketju näyttää siltä, että otoksia olisi enemmän. Tämä toki nähdään myös  $\hat{n}_{eff}$  estimaatista.



Kuva 4.1: Esimerkin regression tulos, vasemmalla *Solar.R* -akselilla ja oikealla *Wind*. Punainen viiva on posteriori keskiarvolla piiretty regressio viiva ja vihreä on vertailun vuoksi PNS mentelmällä sovitettu suora. Siniset ovat sovitteita parametrien eri ehdotus arvoilla.



Kuva 4.2: Esimerkin ketjut ja parametrien posteriori tiheydet



# Luku 5

## Loppusanat

Olemme nyt tarkastelleet perus ajatuksia MCMC-menetelmistä. Kävimme läpi hieman teoriaa, esittelimme algoritmit ja kokeilimme sellaista käytännön tilanteessa. Kuitenkin olemme vain raapaisseet pintaa. Käyttämämme algoritmit ovat monessa mielessä erittäin rajoittuneita. Gibbsin otanta-algoritmissa meidän tulee tuntea marginaalijakaumat, ja Metropolis–Hastings voi olla välillä hidas konvergoitumaan.

Tärkeitä laajennuksia, jotka korjaavat edellisessä kappaleessa mainitsemani ongelmat, on myöhemmin kehitetyt tehokkaammat MCMC-metodit. Tärkeimpänä *Hamiltonian monte carlo* eli HMC. Se pyrkii vähentämään tarvittavien otosten määrää, joka tarvitaan tarpeeksi tarkan posteriori estimaatin saamiseksi, sillä menetelmä vähentää autokorrelaatiota tilojen välillä.

Ohitimme myös useita asioita liittyen Markovin ketjuihin. Markovin ketjujen matematiikka ei ole kovin vaikeaa käytännössä, mutta monet todistukset ovat työläitä ja vaativat matematiikkaa, joka ylittää tämän tutkielman laajuuden.

Kuitenkin MCMC-menetelmät ovat mullistaneet maailmaa muunmuassa koneoppimis menetelmien massakaupallistumisen myötä. Ne ovat myös tehneet tieteestä luotettavampaa mahdollistamalla Bayesiläisen tilastotieteen uuden nousun. Varsinkin replikaatiokriisin jälkeisessä maailmassa se on arvokas kontribuutio.

Paljon on kuitenkin vielä tutkittavaa. Suurista viimeaikaisista loikista huolimatta, MCMC-menetelmä eivät voita vieläkään nopeudessa perinteisiä tilastollisia menetelmiä, joten uusia ja nopeampia algoritmeja olisi vielä kehiteltävä ja helppokäyttöisyyttä parannettava, jotta MCMC-menetelmät saavuttavat täyden potentiaalinsa.

# Kirjallisuus

- [1] Andrew Gelman. *Bayesian Data Analysis*. 3. painos. ISBN: 978-1-4398-4095-5.
- [2] Donald Rubin Gelman Andrew. “Inference from Iterative Simulation using Multiple Sequences”. *Statistical Science* 7 (1992), s. 457–511.
- [3] Stephen Brooks Gelman Andrew. “General Methods for Monitoring Convergence of Iterative Simulations”. *Journal of Computational and Graphical Statistics* (joulukuu 1998). URL: <http://www.stat.columbia.edu/~gelman/research/published/brooksgelman2.pdf>.
- [4] Charles Geyer. “Practical Markov Chain Monte Carlo”. *Statistical Science* 7.4 (1992), s. 473–483.
- [5] Petri Koistinen. *Computational Statistics*. 2009.
- [6] Nicholas Metropolis. “Equation of state calculations by fast computing machines” (1953).
- [7] Christian P. Robert ja George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN: 0387212396.
- [8] Aki Vehtari et al. “Rank-normalization, folding, and localization: An improved  $\hat{R}$  for assessing convergence of MCMC”. *arXiv e-prints*, arXiv:1903.08008 (maaliskuu 2019), arXiv:1903.08008. arXiv: 1903.08008 [stat.CO].