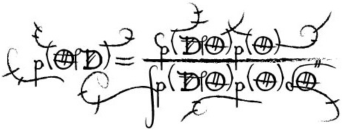

Bayesian Learning 732A46: Lecture 9

Matias Quiroz[1,2]

[1]Division of Statistics and Machine Learning, Linköping University

[2]Research Division, Sveriges Riksbank

May 2016

Lecture overview

- The Metropolis sampler
- The Metropolis-Hastings sampler
- Metropolis-Hastings within Gibbs sampler
- Why does MCMC work?
- Measures of efficiency
- Assessing converge of MCMC simulation

The general idea from last week

- Construct a **Markov sequence** with the property

$$\{\theta^{(i)}\}_{i \geq J}^{N} \text{ is distributed according to } \pi(\theta) \textbf{ for large enough } J.$$

- The posterior $\pi(\theta)$ is the **stationary distribution** of the Markov chain.
- The period $0, 1, \ldots J$ is the **burn-in period** of the chain.
- The draws obtained are used for computing the **expectation of a function**. **Average a function over the posterior distribution**.
- Even if the draws are dependent, still true that

$$\frac{1}{N}\sum_{i=1}^{N} h(\theta^{(i)}) \xrightarrow{a.s} E[h(\theta)] = \int h(\theta)\pi(\theta)d\theta.$$

The Metropolis algorithm

- **Powerful** when distributions are **not of known form**, not even after conditioning.
- A **Markov chain version** of **rejection sampling**.
- **Only requirement**: $\pi(\theta)$ can be **evaluated** (up to $\propto$)
- The **Metropolis** requires a symmetric proposal distribution.
- **Metropolis-Hastings**: **relaxes** the symmetry requirement.

The Metropolis algorithm

The Metropolis algorithm

Obtain N samples from $\pi(\theta) \propto p(y|\theta)p(\theta)$.

- Set an (arbitrary) start point

$$\theta_c = \theta^{(0)},$$

  where θ_c denotes **the current state** of the chain.

- **For** $i = 1, \ldots, N$, **repeat**

1. **Propose** a draw $\theta_p \sim q(\theta|\theta_c)$ (q - proposal distribution).
2. Evaluate

$$\alpha(\theta_c, \theta_p) = \min\left(1, \frac{\pi(\theta_p)}{\pi(\theta_c)}\right) = \min\left(1, \frac{p(y|\theta_p)p(\theta_p)}{p(y|\theta_c)p(\theta_c)}\right).$$

3. Sample $u \sim \mathrm{uniform}(0, 1)$.
4. If $u \leq \alpha(\theta_c, \theta_p) \implies \theta^{(i)} = \theta_p$, else $\theta^{(i)} = \theta_c$

The Metropolis algorithm, cont

- **"Climbing up the hill"** will always be accepted.
- **"Down the hill"** accepted with fraction $\pi(\theta_p)/\pi(\theta_c)$.
- **Note**: if we reject the draw we **keep the current draw in the chain**. A Metropolis that rejects **too often** gives a "sticky" chain.
- **Common choice of proposal**: $q(\cdot|\theta_c) = \mathcal{N}(\theta_c, \Sigma)$ (has to be symmetric). **Random walk** type (notice the mean).
- $\Sigma = \tilde{c}I$. Choose $\tilde{c}$ so that your **acceptance probability** (on average) is $\alpha \approx 0.23$.
- If the parameters are **heavily correlated**: $\Sigma = \tilde{c}\Sigma_{\theta^\star}$, where $\Sigma_{\theta^\star}$ is the **posterior covariance** evaluated at the mode $\theta^\star$ (**recall**: `optim` in `R`).
- **Question:** Why do you think that $\alpha \approx 1$ is **not** desirable with a Random walk proposal?

The Metropolis algorithm, cont

- **"Climbing up the hill"** will always be accepted.
- **"Down the hill"** accepted with fraction $\pi(\theta_p)/\pi(\theta_c)$.
- **Note**: if we reject the draw we **keep the current draw in the chain**. A Metropolis that rejects **too often** gives a "sticky" chain.
- **Common choice of proposal**: $q(\cdot|\theta_c) = \mathcal{N}(\theta_c, \Sigma)$ (has to be symmetric). **Random walk** type (notice the mean).
- $\Sigma = \tilde{c}I$. Choose $\tilde{c}$ so that your **acceptance probability** (on average) is $\alpha \approx 0.23$.
- If the parameters are **heavily correlated**: $\Sigma = \tilde{c}\Sigma_{\theta^\star}$, where $\Sigma_{\theta^\star}$ is the **posterior covariance** evaluated at the mode $\theta^\star$ (**recall**: `optim` in `R`).
- **Question:** Why do you think that $\alpha \approx 1$ is **not** desirable with a Random walk proposal?
- Will give a **slow mixing** (**inefficient**) chain. More on this later.

The Metropolis-Hastings algorithm

- A **more general** version of the **Metropolis algorithm**.
- **Same setting**: we can evaluate

$$\pi(\theta) \propto p(y|\theta)p(\theta).$$

- **Metropolis-Hastings**: Symmetry of proposal is not required.
- **What do we gain?**: can move away from a **Random Walk** (RW) $q()$.
- **Note**:
  The RW proposal is **local** (proposes from the **current state** of the chain).
  Moves around slowly in θ space.
- A **good proposal** $q()$ explores the parameter space **efficiently**.
  Propose globally (where the posterior mass is located).

The Metropolis-Hastings algorithm, cont.

The Metropolis-Hastings algorithm

Obtain N samples from $\pi(\theta) \propto p(y|\theta)p(\theta)$.

- Set an (arbitrary) start point

$$\theta_c = \theta^{(0)},$$

  where θ_c denotes **the current state** of the chain.

- **For** $i = 1, \ldots, N$, **repeat**

1. **Propose** a draw $\theta_p \sim q(\theta|\theta_c)$ (q - proposal distribution).
2. Evaluate

$$\alpha(\theta_c, \theta_p) = \min\left(1, \frac{\pi(\theta_p)/q(\theta_p|\theta_c)}{\pi(\theta_c)/q(\theta_c|\theta_p)}\right) = \min\left(1, \frac{p(y|\theta_p)p(\theta_p)/q(\theta_p|\theta_c)}{p(y|\theta_c)p(\theta_c)/q(\theta_c|\theta_p)}\right).$$

3. Sample $u \sim \mathrm{uniform}(0, 1)$.
4. If $u \leq \alpha(\theta_c, \theta_p) \implies \theta^{(i)} = \theta_p$, else $\theta^{(i)} = \theta_c$

The Metropolis-Hastings algorithm, cont

- **Note**: if $q(\theta) =$ symmetric $\implies$ **Metropolis** algorithm [q cancels].
- **Independence Metropolis-Hastings**:

$$q(\theta|\theta_c) = q(\theta) \quad \text{[ind. of the current state (\textbf{not a RW})]}$$

- **Example**:

$$q(\cdot) = t_\nu(\theta^\star, \Sigma_{\theta^\star}),$$

where

$$\begin{aligned} \theta^\star &= \text{the \textbf{mode} from a numerical optimization} \\ \Sigma_{\theta^\star} &= \text{the \textbf{covariance} at } \theta^\star \quad [-H_{\theta^\star}^{-1}]. \end{aligned}$$

- Very efficient... **but can get stuck**!
- Make sure $t_\nu(\theta^\star, \Sigma_{\theta^\star})$ has **heavier tails** than $p(y|\theta)p(\theta)$.

Metropolis-Hastings within Gibbs algorithm

- **Recall Gibbs**: sample the **blocks** $\theta = (\theta_1, \ldots, \theta_K)$., by

$$\pi(\theta_1|\theta_2, \theta_3 \ldots, \theta_K)$$

$$\vdots$$

$$\pi(\theta_K|\theta_1, \theta_2, \ldots, \theta_{K-1})$$

- **Assumption**: can sample from each **full conditional** [**known form**].
- **What if** not all are of known form? **M-H within Gibbs** to the rescue.
- **Example:** let $K = 3$ and suppose $\pi(\theta_2|\theta_1, \theta_3)$ is **not** of known form.

  Updating θ_2 **at iteration** i: **Propose**

$$\theta_p = \theta_2^{(i)} \sim q(\theta_2|\theta_1^{(i)}, \theta_2^{(i-1)}, \theta_3^{(i-1)}) \quad \left[\textbf{Note}: \theta_c = \theta_2^{(i-1)}\right].$$

  Then

$$\alpha = \min\left(1, \frac{\pi(\theta_p|\theta_1^{(i)}, \theta_3^{(i-1)})/q(\theta_p|\theta_1^{(i)}, \theta_c, \theta_3^{(i-1)})}{\pi(\theta_c|\theta_1^{(i)}, \theta_3^{(i-1)})/q(\theta_c|\theta_1^{(i)}, \theta_p, \theta_3^{(i-1)})}\right), \textbf{ decide} \text{ to accept/reject.}$$

Heterosedastic regression

- **M-H within Gibbs: Heteroscedastic regression**:

$$y_i = x_i'\beta + \varepsilon_i$$

  where the errors are **heteroscedastic**

$$\varepsilon_i \sim \mathcal{N}\left(0, \sigma^2 \exp\left(x_i'\gamma\right)\right).$$

- **Priors**:
  - **Multivariate normal** for β and γ.
  - Inv-χ^2 for σ^2.
- **Gibbs sampling (two blocks)**:
  - $\beta, \sigma^2 | \gamma, y$
  - $\gamma | \beta, \sigma^2, y$

M-H within Gibbs: Heteroscedastic regression, cont.

- Draws from $\beta, \sigma^2|\gamma, y$ can be obtained as in standard (**homoscedastic**) linear regression but on **transformed data**. **Standard trick**.
- Rewrite the model as
$$\tilde{y}_i = \tilde{x}_i'\beta + \tilde{\varepsilon}_i,$$
where
  - $\tilde{y}_i = \exp\left(-x_i'\gamma/2\right) y_i$
  - $\tilde{x}_i' = \exp\left(-x_i'\gamma/2\right) x_i'$
  - $\tilde{\varepsilon}_i = \exp\left(-x_i'\gamma/2\right) \varepsilon_i$.
  - Note that $Var(\tilde{\varepsilon}_i) = \sigma^2$, so **homoscedastic**.
- $p(\beta, \sigma^2|\gamma, y)$ - using a $\mathcal{N}$-Inv-χ^2 conjugate prior (with transformed data).
- $p(\gamma|\beta, \sigma^2, y)$ is non-standard, but we can use **M-H to sample** with a Random walk proposal...
- ... Or an **independence M-H proposal** $\mathcal{N}(\gamma^\star, \Sigma_{\gamma^\star})$, $\gamma^\star, \Sigma_{\gamma^\star}$ obtained with `optim` in `R`.

Connecting the Gibbs sampler to M-H

- **Updating a block** in a **Gibbs** step is a **special case** of M-H where

$$\textbf{Proposal} = \textbf{Full conditional posterior},$$

  so that $\alpha = 1$.
- **In our example**

$$q(\theta_2|\theta_1^{(i)}, \theta_2^{(i-1)}, \theta_3^{(i-1)}) = \pi(\theta_2|\theta_1^{(i)}, \theta_3^{(i-1)}) \quad [\textbf{gives } \alpha = 1] \,.$$

Why does MCMC work?

- **Excellent paper**: Chib and Greenberg (1995).
- The **transition kernel** of the M-H Markov chain:

$$T(\theta_c \to d\theta_p) = \overbrace{\int T(\theta_c \to \theta_p) d\theta_p}^{\text{Pr(move)}} + \overbrace{r(\theta_c)}^{\text{Pr(stay)}} \delta_{\theta_c}(d\theta_p),$$

where

$$T(\theta_c \to \theta_p) = q(\theta_p|\theta_c)\alpha(\theta_c, \theta_p) \quad \text{and } r(\theta_c) = 1 - \int T(\theta_c \to \theta_p) d\theta_p,$$

with

$$\delta_{\theta_c}(d\theta_p) = \begin{cases} 1, & \text{if } \theta_c \in d\theta_p \\ 0, & \text{if } \theta_c \notin d\theta_p. \end{cases}$$

- **M-H** chooses α so that

$$\pi(\theta_c) T(\theta_c \to \theta_p) = \pi(\theta_p) T(\theta_p \to \theta_c) \quad \text{[\textbf{detailed balance}]}.$$

Why does MCMC work?, cont.

Proof that M-H's transition kernel fulfills detailed balance [extra, if you are interested].

$$\left[\alpha(\theta_c, \theta_p) = \min\left(1, \frac{\pi(\theta_p)/q(\theta_p|\theta_c)}{\pi(\theta_c)/q(\theta_c|\theta_p)}\right) \quad \text{and} \quad T(\theta_c \to \theta_p) = q(\theta_p|\theta_c)\alpha(\theta_c, \theta_p)\right]$$

$$\begin{aligned}
\pi(\theta_c)T(\theta_c \to \theta_p) &= \pi(\theta_c)q(\theta_p|\theta_c)\min\left(1, \frac{\pi(\theta_p)/q(\theta_p|\theta_c)}{\pi(\theta_c)/q(\theta_c|\theta_p)}\right) \\
&= \pi(\theta_c)q(\theta_p|\theta_c)\min\left(\frac{\pi(\theta_c)q(\theta_p|\theta_c)}{\pi(\theta_c)q(\theta_p|\theta_c)}, \frac{\pi(\theta_p)q(\theta_c|\theta_p)}{\pi(\theta_c)q(\theta_p|\theta_c)}\right) \\
&= \pi(\theta_p)q(\theta_c|\theta_p)\min\left(\frac{\pi(\theta_c)q(\theta_p|\theta_c)}{\pi(\theta_p)q(\theta_c|\theta_p)}, 1\right) \\
&= \pi(\theta_p)q(\theta_c|\theta_p)\alpha(\theta_p, \theta_c) \\
&= \pi(\theta_p)T(\theta_p \to \theta_c).
\end{aligned}$$

Thus, $\pi(\theta)$ is the **stationary distribution** of the Markov chain generated by **M-H**.

□

- **Convergence to** $\pi(\theta)$: q has **positive density** on the support of $\pi(\theta)$.

Illustrating the concept of efficiency

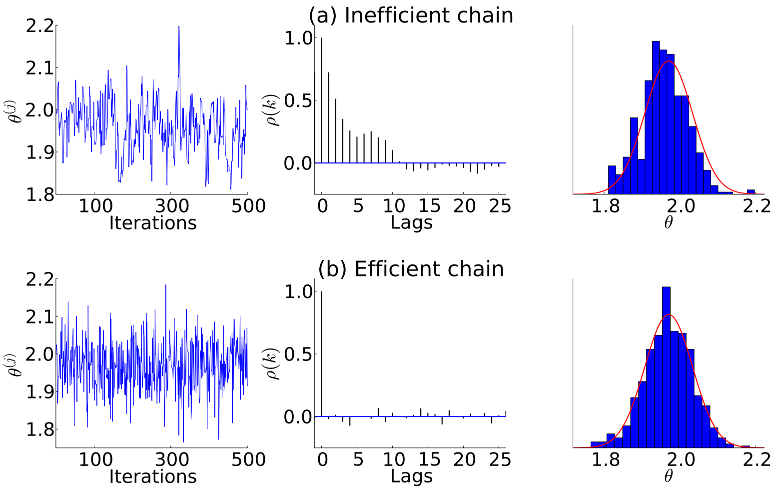

Figure : **Left:** trace plots of chain. **Middle:** auto-correlation of chain at lag k. **Right:** True posterior (red line) and MCMC approximation (histogram)

Measures of efficiency - IF and ESS

- With **MCMC**: The generated $\{\theta^{(i)}\}_{i=1}^{N}$ is a **dependent** sequence.
- How **efficient** is **MCMC** compared to **iid. sampling**?
- **Variance of posterior mean estimate** if the **sequence is iid**.

$$V[\bar{\theta}] = Var\left[\frac{1}{N}\sum_{i=1}^{N}\theta^{(i)}\right] = \frac{\sigma^2}{N} \quad \left[\sigma^2 = V\left[\theta\right]\right].$$

- **Variance of posterior mean estimate** if the **sequence is dependent**

$$V[\bar{\theta}] = Var\left[\frac{1}{N}\sum_{i=1}^{N}\theta^{(i)}\right] = \frac{\sigma^2}{N} \times IF, \quad IF = \left(1 + 2\sum_{k=1}^{\infty}\rho_k\right),$$

where $\rho_k = Corr(\theta^{(i)}, \theta^{(i+k)})$ is the **auto-correlation** at lag k.

Measures of efficiency - IF and ESS, cont.

- IF is the **Inefficiency Factor** (IF) (or *integrated auto-correlation time*): The variance of the estimate **inflates** IF times for my MCMC (relative to iid. sampling).

- **Effective Sample Size** (ESS): **ESS** $= N/IF$.

- **Tells you**: how many **equivalent to iid. draws** you get with your MCMC.

- Can be computed with the `CODA` package in `R` (Plummer et al., 2006). **Useful function**: `effectiveSize`.

Improving the efficiency of MCMC

- Most **essential** (but also the **most difficult**) find a **better proposal** q.
- **Modify** your proposal.
  For example in a **R-W Metropolis** make sure $\alpha \approx 0.23$.

  $$\tilde{c} = \frac{2.4}{\sqrt{p}} \text{ gives [\textbf{in theory}] } \alpha \approx 0.23 \quad [p = \text{number of parameters}].$$

  Note: only for a **R-W**. With **IMH** you want α as high as possible.
- **Re-parametrization** helps a lot. **Especially** if the support of θ is **restricted**.
- **Example**

  $$\begin{aligned} &\text{if } \theta \in \mathbb{R}^+ \quad \text{use} \quad \phi = \log(\theta) \\ &\text{if } \theta \in [0,1] \quad \text{use} \quad \phi = \text{logit}(\theta), \end{aligned}$$

  but (**again**!) **do not forget the Jacobian**! [transformation of variables]
- Simple way to **reduce** auto-correlation: **thinning** - keep every bth sample.

Assessing convergence of MCMC

- **How long** is the **burn-in** period?
- **Convergence diagnostics**:
  - **Plot the Markov chains**. Do they seem to settle?
  - **Plot cumulative means**. Do the means converge?
  - Interested in a function $h(\theta)$? **Monitor its convergence**.
    Example:
    Objective: $h(\theta) = \Pr(\theta > 2)$. **MCMC estimate** is

$$\hat{I}_N = \{\#\{\theta^{(i)}\}_{i=1}^{N} > 2\}/N \quad \text{[when } \textbf{all } N \text{ draws are available].}$$

    Compute (and plot) $\hat{I}_k$ for $k = 1, \ldots, N$ and see if it converges.
  - **Do you suspect** your posterior is **multimodal**? Try different starting values.
- **Question**: How long to sample **after** the **burn-in** period?
- **Answer**: depends on IF (and ESS). An ESS of 1000 is usually sufficient for most tasks.

References

Chib, S. and Greenberg, E. (1995). Understanding the M-H algorithm. *The American Statistician*, 49(4):327-335.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC, *R News*, 6(1):7-11.