



STK4051/STK9051

Computational Statistics

Riccardo De Bin

`debin@math.uio.no`

Outline of the lecture

- Review
 - Markov Chains
- Markov Chain Monte Carlo
 - Metropolis–Hastings Algorithm
 - Independence Chains
 - Random Walk Chains

Markov Chains: Definitions

Remember that:

- a discrete-time, discrete-space **Markov chain** is $X^{(0)}, X^{(1)}, \dots$ where $X^{(t)}$ obeys the **Markov property** that

$$P \left[X^{(t)} \middle| x^{(0)}, \dots, x^{(t-1)} \right] = P \left[X^{(t)} \middle| x^{(t-1)} \right];$$

- a Markov chain is governed by a **transition probability matrix**, $\mathbf{P}$, with the ij -th element being

$$P_{ij}^{(t)} = P \left[X^{(t+1)} = j \middle| X^{(t)} = i \right];$$

- Let π denote a vector of probabilities with the i -th element π_i denoting the **marginal probability** that $X^{(t)} = i$. Then the **marginal distribution** of $X^{(t+1)}$ must be $\pi^T \mathbf{P}$;
- if $\pi^T \mathbf{P} = \pi^T$ then π is a **stationary distribution** for $\mathbf{P}$.

Markov Chains: Definitions

Moreover:

- a Markov chain is **irreducible** if **any** state j can be reached from **any** state i in a **finite number** of steps for all i and j ,

$$\forall i, j, \exists m > 0 : P[X^{(m+n)} = i | X^{(n)} = j] > 0;$$

- a Markov chain is **periodic** if it can visit certain portions of the state space only at **regularly spaced intervals**;
- state j has **period** d if the probability of going from state j to state j in n steps is 0 for all n **not divisible** by d ;
- if every state in a Markov chain has **period 1**, then the chain is **aperiodic**.

Markov Chains: Stationary distribution

If $X^{(1)}, X^{(2)}, \dots$ are realizations from an **irreducible and aperiodic Markov chain** with stationary distribution π , then:

- $X^{(n)}$ **converges in distribution** to the distribution given by π ;
- for any function h ,

$$\frac{1}{n} \sum_{t=1}^n h(X^{(t)}) \rightarrow E_{\pi}\{h(X)\}$$

almost surely as $n \rightarrow \infty$, provided $E_{\pi}\{|h(X)|\}$ exists.

This is one form of the **ergodic theorem**, which is a generalization of the **strong law of large numbers**.

Markov Chain Monte Carlo: Introduction

Markov chain Monte Carlo (MCMC) methods:

- can be used to **generate a draw** from a distribution that **approximates** some target distribution f ,
 - ▶ usually f is the posterior distribution (Bayesian paradigm);
- more properly viewed as methods for generating a sample from which **expectations of functions** of $\mathbf{X}$ can be estimated;
- the art of MCMC lies in the construction of a **suitable chain**:
 - ▶ the distribution of $X^{(t)}$ may differ substantially from f ;
 - ▶ $X^{(t)}$ are serially dependent.

Markov Chain Monte Carlo: Introduction

Let $\{\mathbf{X}^{(t)}\}$ denote a Markov chain for $t = 0, 1, 2, \dots$, where $\mathbf{X}^{(t)} = (X_1^{(t)}, \dots, X_p^{(t)})$, with state space $\mathcal{S}$ either continuous or discrete:

- the MCMC sampling strategy is to **construct an irreducible, aperiodic Markov chain** for which the stationary distribution equals the target distribution, f ;
- for sufficiently large t , a realization, $\mathbf{X}^{(t)}$, from this chain will have **approximate marginal distribution** f ;
- there are **many strategies** for constructing a MCMC algorithm:
 - ▶ the Metropolis-Hastings algorithm;
 - ▶ the Gibbs sampler;
 - ▶ ...

Metropolis–Hastings Algorithm: introduction

The method begins at $t = 0$ with the selection of $\mathbf{X}^{(0)} = \mathbf{x}^{(0)}$,

- drawn at random from some starting distribution, g ;
- with the requirement that $f(\mathbf{x}^{(0)}) > 0$.

Given $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$, the algorithm generates $\mathbf{X}^{(t+1)}$ as follows:

1. sample a candidate $\mathbf{X}^*$ from a proposal distribution $g(\cdot | \mathbf{x}^{(t)})$;
2. compute the **Metropolis-Hastings ratio**,

$$R(\mathbf{x}^{(t)}, \mathbf{X}^*) = \frac{f(\mathbf{X}^*) g(\mathbf{x}^{(t)} | \mathbf{X}^*)}{f(\mathbf{x}^{(t)}) g(\mathbf{X}^* | \mathbf{x}^{(t)})};$$

3. sample a value for $\mathbf{X}^{(t+1)}$ according to:

$$\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{X}^* & \text{with probability } \min\{R(\mathbf{x}^{(t)}, \mathbf{X}^*), 1\} \\ \mathbf{x}^{(t)} & \text{otherwise} \end{cases};$$

4. increment t and return to step 1.

Metropolis–Hastings Algorithm: remarks

Remarks:

- if the chain is **irreducible** and **aperiodic**,
 - ▶ the **unique limiting stationary distribution** of the chain from the Metropolis-Hastings algorithm is the target distribution;
 - ▶ irreducibility and aperiodicity **depends on** the proposal distribution;
 - ▶ they **must be checked** diligently for any implementation.
- the sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ will likely include **multiple copies of some points** in the state space,
 - ▶ occurs when $\mathbf{X}^{(t+1)}$ **retains** the previous value $\mathbf{x}^{(t)}$ rather than jumping to the proposed value $\mathbf{x}^*$;
 - ▶ it is important to **keep** these copies;
 - ▶ the **frequencies** of sampled points are used to correct for the fact that the proposal density differs from the target density.

Metropolis–Hastings Algorithm: remarks

Remarks:

- persistent **dependence** of the chain **on its starting value** can seriously degrade its performance
 - ▶ **omit some of the initial realizations** of the chain;
 - ▶ it is called the **burn-in period**;
 - ▶ it is an **essential** component of MCMC applications;
- it is also a good idea to run MCMC procedures like the Metropolis–Hastings algorithm from **multiple starting points**,
 - ▶ to check for **consistent** results;
 - ▶ as seen for optimization algorithms;
- more on these issues in two weeks (§7.3).

Metropolis–Hastings Algorithm: choosing a proposal distribution

A well chosen proposal distribution produces:

- candidates that **cover the support** of the stationary distribution in a reasonable number of iterations;
- candidates that are not accepted or rejected **too frequently**.

Both points are related to the spread of the proposal distribution,

- if **too diffuse**:
 - ▶ candidates are **rejected frequently**;
 - ▶ the chain requires **many iterations** to adequately **explore** the space of the target distribution;
- if **too focused**:
 - ▶ the chain **remains in one small region** for many iterations;
 - ▶ other regions of the target distribution are **not** adequately explored.

Metropolis–Hastings Algorithm: expectations of functions

The distribution of realizations from the Metropolis-Hastings chain **approximates** the stationary distribution of the chain as $t \rightarrow \infty$, so

$$E \{h(\mathbf{X})\} \approx \frac{1}{n} \sum_{i=1}^n h \left(\mathbf{x}^{(i)} \right).$$

Some useful quantities that can be estimated include:

- means, $E \{h(\mathbf{X})\}$;
- variances, $E \{ (h(\mathbf{X}) - E \{h(\mathbf{X})\})^2 \}$;
- tail probabilities, $E \{ 1_{\{h(\mathbf{X}) \leq q\}} \}$ for a constant q ;
- virtually any other statistic that may be of interest!

It can also estimate f itself using density estimation methods.

Metropolis–Hastings Algorithm: Independence chains

Suppose that the proposal distribution for the Metropolis-Hastings algorithm is chosen such that $g(\mathbf{x}^* | \mathbf{x}^{(t)}) = g(\mathbf{x}^*)$:

- this yields an **independence chain**, where each candidate is drawn **independently** of the past;
- the Metropolis-Hastings ratio is then

$$R(\mathbf{x}^{(t)}, \mathbf{X}^*) = \frac{f(\mathbf{X}^*) g(\mathbf{x}^{(t)})}{f(\mathbf{x}^{(t)}) g(\mathbf{X}^*)};$$

- the resulting Markov chain is **irreducible** and **aperiodic** if $g(\mathbf{x}) > 0$ whenever $f(\mathbf{x}) > 0$;
- $\frac{f(\mathbf{x}^*)g(\mathbf{x}^{(t)})}{f(\mathbf{x}^{(t)})g(\mathbf{x}^*)} = \frac{f(\mathbf{x}^*)/g(\mathbf{x}^*)}{f(\mathbf{x}^{(t)})/g(\mathbf{x}^{(t)})}$ is a ratio of **importance weights**,
 - ▶ the algorithm behaves like **importance sampling** and **SIR**;
 - ▶ if $f(\mathbf{x}^{(t)})/g(\mathbf{x}^{(t)}) \gg f(\mathbf{x}^*)/g(\mathbf{x}^*)$, the chain tends to **get stuck** for long periods at the current value.

Metropolis–Hastings Algorithm: Bayesian inference and MCMC

The Bayesian paradigm is based on the **Bayes theorem**,

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})}{p(\mathbf{y})} \propto p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta}),$$

where

- $p(\boldsymbol{\theta}|\mathbf{y})$ is the **posterior density** of $\boldsymbol{\theta}$;
- $p(\mathbf{y}|\boldsymbol{\theta})$ is the **likelihood**;
- $p(\boldsymbol{\theta})$ is the **prior density**.

In Bayesian applications one uses MCMC to simulate from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$,

- i.e., the **target distribution is the posterior distribution**.

Metropolis–Hastings Algorithm: Bayesian inference and MCMC

Simple strategy: use the **prior as a proposal distribution** in an independence chain.

In our Metropolis-Hastings notation,

$$f(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}) \quad \text{and} \quad g(\boldsymbol{\theta}^*) = p(\boldsymbol{\theta}^*).$$

Then the Metropolis-Hastings ratio is given by

$$R(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^*) = \frac{p(\mathbf{y}|\boldsymbol{\theta}^*)}{p(\mathbf{y}|\boldsymbol{\theta}^{(t)})}.$$

Remarks:

- if the proposal distribution **is equal to** the prior distribution,
 - the Metropolis-Hastings ratio equals the **likelihood ratio**;
- the support of the prior **covers** the support of the posterior,
 - the stationary distribution of this chain is the desired posterior;
- more sophisticated algorithms often have better performance.

Metropolis–Hastings Algorithm: Mixture Distribution Example

Suppose we have observed data $y_1, y_2, \dots, y_{100}$,

- sampled independently from the mixture distribution,

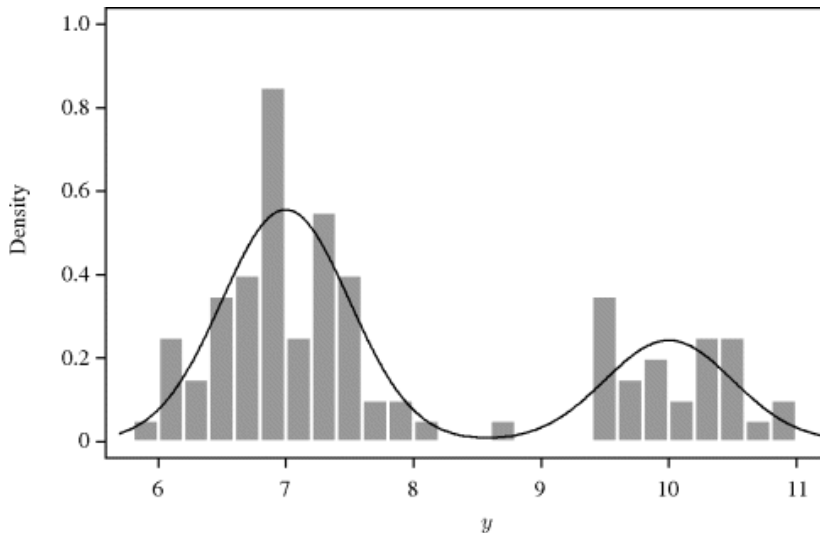
$$\delta N(7, 0.5^2) + (1 - \delta)N(10, 0.5^2);$$

- histogram in the next slides, generated with $\delta = 0.7$.

Note:

- mixture densities are **common** in real-life applications,
 - ▶ e.g., the data may come from more than one population;
- goal: use MCMC techniques to construct a chain,
 - ▶ whose stationary distribution equals the posterior density of δ ;
 - ▶ assuming a $\text{Unif}(0, 1)$ prior distribution for δ .

Metropolis–Hastings Algorithm: Mixture Distribution Example



(Givens & Hoeting, 2013, figure 7.1)

Metropolis–Hastings Algorithm: Mixture Distribution Example

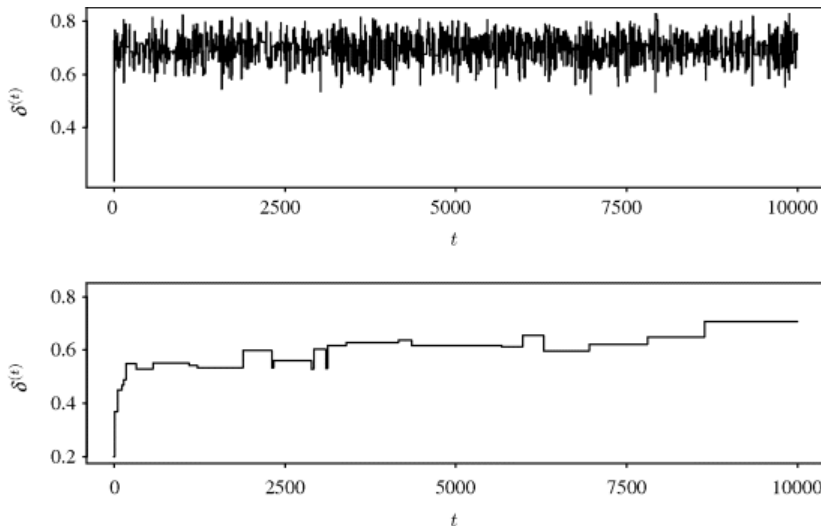
Try two different MCMC proposal densities:

- Beta(1, 1): equivalent to a Unif(0, 1) distribution;
- Beta(2, 10):
 - ▶ skewed right with mean of 0.167;
 - ▶ values of δ near 0.7 are unlikely to be generated from the proposal distribution.

Results using 10000 iterations of Metropolis-Hastings algorithm for each proposal, reported in terms of:

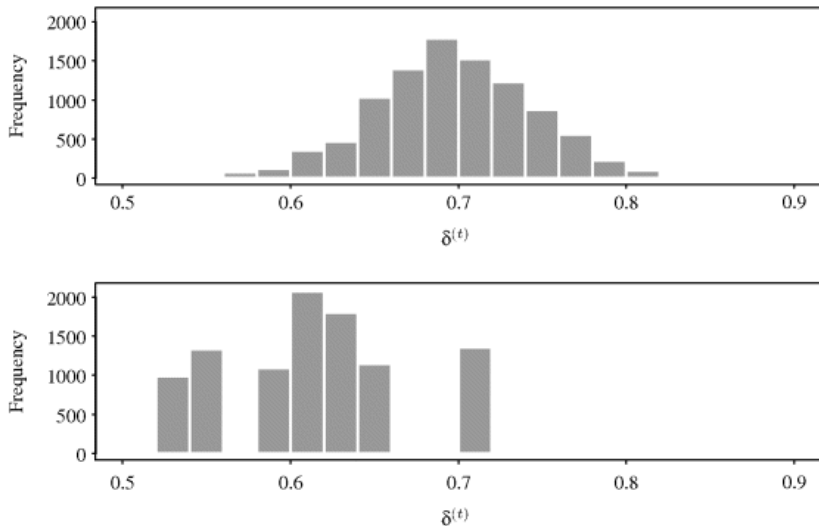
- **sample paths**,
 - ▶ a plot of chain realizations, $\delta^{(t)}$, vs the iteration number, t ;
 - ▶ useful for investigating the behavior of the Markov chain;
- **histogram of realizations**.

Metropolis–Hastings Algorithm: Mixture Distribution Example



(Givens & Hoeting, 2013, figure 7.2)

Metropolis–Hastings Algorithm: Mixture Distribution Example



(Givens & Hoeting, 2013, figure 7.3)

Metropolis–Hastings Algorithm: Random walk chains

A **random walk chain** is a Markov chain produced via a **simple variant** of the Metropolis–Hastings algorithm:

- draw $\epsilon \sim h(\epsilon)$ for some density h ;
- set $\mathbf{X}^* = \mathbf{x}^{(t)} + \epsilon$.

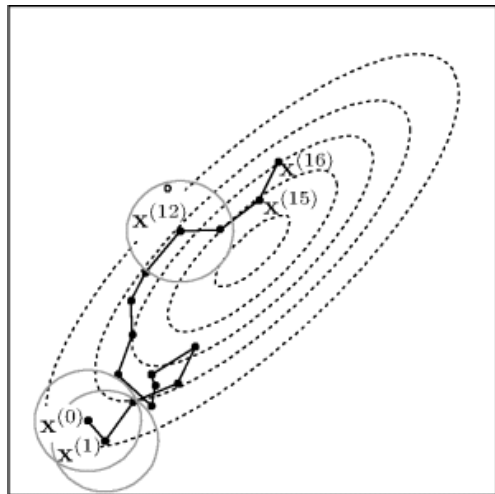
Note that the **proposal** is then given by

$$g(\mathbf{x}^* | \mathbf{x}^{(t)}) = h(\mathbf{x}^* - \mathbf{x}^{(t)}).$$

Common choices for h :

- uniform distribution over a ball centred at the origin;
- scaled standard normal distribution;
- scaled Student's t distribution.

Metropolis–Hastings Algorithm: Random walk chains



(Givens & Hoeting, 2013, figure 7.4)

Hypothetical random walk chain for sampling a two dimensional target distribution (dotted lines) using proposed increments sampled uniformly from a disk centred at the current value.

Metropolis–Hastings Algorithm: Mixture Distribution Example

Back to the Mixture Distribution Example,

- consider using a random walk chain to learn about the posterior for δ under a $\text{Unif}(0, 1)$ prior.

We generate proposals by adding a $\text{Unif}(-a, a)$ to the current $\delta^{(t)}$:

- it is likely that some proposals will be generated **outside** the interval $[0, 1]$ during the progression of the chain;
- **inelegant** approach: note that the posterior is 0 $\forall \delta \notin [0, 1]$,
 - ▶ thereby **forbidding** steps to such points;
- better approach: **re-parametrise** the problem:
 - ▶ let $U = \text{logit}\{\delta\} = \log\{\delta/(1 - \delta)\}$;
 - ▶ run a random walk chain **on** U ;
 - ▶ generate a proposal by adding a $\text{Unif}(-b, b)$ to $u^{(t)}$.

Metropolis–Hastings Algorithm: Mixture Distribution Example

There are two ways to view the reparameterization.

The first option is to **run the chain in δ -space**:

- the **proposal density** $g(\cdot|u^{(t)})$ is transformed into a **proposal density** in δ -space, taking account of the Jacobian;
- the **Metropolis-Hastings ratio** for a proposed value δ^* is then

$$\frac{f(\delta^*) g(\text{logit}\{\delta^{(t)}\}|\text{logit}\{\delta^*\}) |J(\delta^{(t)})|}{f(\delta^{(t)}) g(\text{logit}\{\delta^*\}|\text{logit}\{\delta^{(t)}\}) |J(\delta^*)|},$$

where $|J(\delta^{(t)})|$ is the absolute value of the determinant of the Jacobian for the transformation from δ to u , evaluated at $\delta^{(t)}$.

Metropolis–Hastings Algorithm: Mixture Distribution Example

The second option is to **run the chain in u -space**:

- the **target density** for δ is transformed into a **target density** for u , taking into account the Jacobian;
- $\delta = \text{logit}^{-1}\{U\} = \exp\{U\}/(1 + \exp\{U\})$;
- for $U^* = u^*$, this yields the **Metropolis-Hasting ratio**

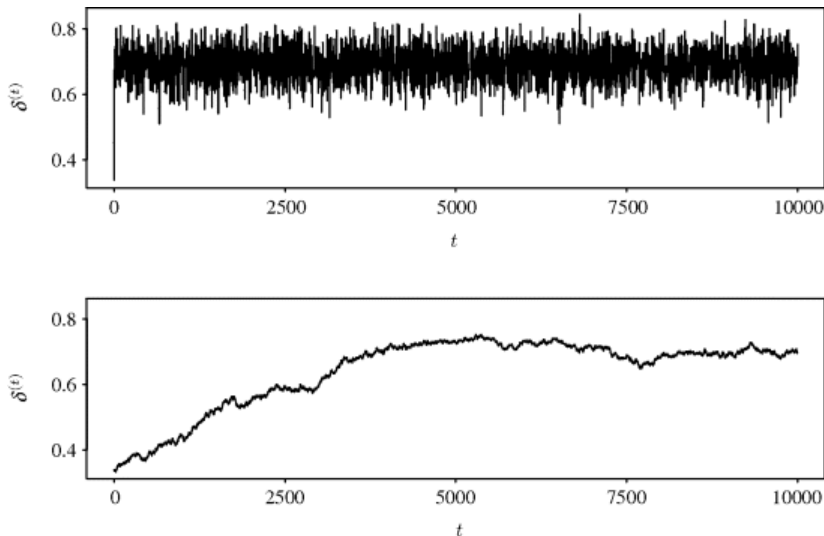
$$\frac{f(\text{logit}^{-1}\{u^*\}) |J(u^*)| g(u^{(t)}|u^*)}{f(\text{logit}^{-1}\{u^{(t)}\}) |J(u^{(t)})| g(u^*|u^{(t)})};$$

- $|J(u^*)| = 1/|J(\delta^*)|$, so the options produce equivalent chains.

Next slide shows sample paths for δ from **two random walk chains** run in u -space, setting $U^* = u^{(t)} + \epsilon$:

- top panel, chain generated by drawing $\epsilon \sim \text{Unif}(-1, 1)$;
- bottom panel, $\epsilon \sim \text{Unif}(-0.01, 0.01)$.

Metropolis–Hastings Algorithm: Mixture Distribution Example



(Givens & Hoeting, 2013, figure 7.5)

References |

GIVENS, G. H. & HOETING, J. A. (2013). *Computational Statistics*. Wiley.