Markov Chain Monte Carlo

Matthew Stephens and Heejung Shim

Learning goals

Metropolis-Hastings Algorithm:

- Understand key ideas for the algorithm
- Be able to derive and implement the algorithm
- (not examinable; challenging) Obtain intuitions on the theory to support why the algorithm works

Gibbs sampler:

- Understand it is a special case of the Metropolis-Hastings Algorithm
- Understand its potential limitation compared to the Metropolis-Hastings Algorithm

Markov chain Monte Carlo (MCMC)

MCMC is a very widely-used technique for simulating samples from the posterior distribution. $p(\theta|x)$

The basic idea is to simulate a Markov chain, whose stationary distribution is the posterior distribution. By simulating the Markov chain for long enough, one obtains samples that are "approximately" from the posterior distribution.

- Metropolis–Hastings Algorithm
- Gibbs Sampling

Metropolis–Hastings Algorithm

eg $\theta' \sim N(\theta, 1^2)$

Let $q(\theta, \theta')$ denote a transition density, which is a way of generating new states θ' given the current state θ.

Let π denote the target distribution, up to a constant of proportionality (e.g. to simulate from the posterior, $\pi = p(\theta)p(x|\theta)$).

$\theta \sim p(\theta|x) \propto p(\theta)\,p(x|\theta)$

Metropolis–Hastings Algorithm

don't know exact posterior distribution

$\theta \sim p(\theta|x) \propto p(\theta)p(x|\theta) = \pi(\theta)$

for $q(\theta, \theta')$: $\theta' \sim N(\theta, 1^2)$

$t=0$ $\theta^{(0)}$

$t=1$ $\theta^{(1)}$

$t=2$ $\theta^{(2)}$

$\vdots$

$t=M$ $\theta^{(M)}$

$M \to \infty$ $\theta^{(M)} \sim p(\theta|x)$

burn in

Now consider the following Markov chain:

1. Start with an initial value $\theta^{(0)}$; set $t = 0$.
2. Given the current value of $\theta^{(t)} = \theta$, generate a proposed new value θ' according to $q(\theta, \cdot)$.
3. Define the acceptance probability A by

$t=0$ $\theta=1$ $\theta' \sim N(1, 1^2)$ $A = \min\left\{1, \frac{\pi(\theta')q(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}\right\}$ constant part will be cancelled out

$\theta' = 0.5$ $t=1$ $\theta = 0.5$ $\theta' \sim N(0.5, 1^2)$ $\theta' = 2$

$$A = \min\left(1, \frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}\right).$$

4. With probability A set $\theta^{(t+1)} = \theta'$; otherwise set $\theta^{(t+1)} = \theta$.

with probability $1-A$

5. Increase t; return to step 2.

Note: q is referred to as the *proposal distribution*. The probability A is the *acceptance probability*.

Why does Metropolis–Hastings algorithm work?

See MarkovProcesses.pdf for definition and detailed explanation.

- The process $\theta^{(t)}$ has stationary distribution π.
  - Proof: check detailed balance condition, $\pi(\theta)p(\theta \to \theta') = \pi(\theta')p(\theta' \to \theta)$.
- π will be limiting.
  - The process $\theta^{(t)}$ can stay where it is, it is aperiodic.
  - For all sensible choices of q it will also be irreducible (i.e., regardless the present state, it can reach any other state in finite time).
  - If a Markov process is aperiodic and irreducible then any stationary distribution is unique and limiting.
- The process is ergodic.
  - The process $\theta^{(t)}$ is aperiodic and irreducible, and π is a limiting distribution, then the process is ergodic.

Why does Metropolis–Hastings algorithm work?

Because the output process $\theta^{(t)}$ has a limiting distribution π, the output of the M-H algorithm will—in the limit—be a sequence of observations from the density π.

$\theta^{(M)} \sim \pi(\theta) = p(\theta|x)$ when $M \to \infty$

These observations are not independent, but because they come from an ergodic process we can still use them to estimate properties of π.

Metropolis–Hastings Algorithm

requirement

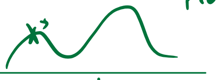
Remarks

1. For this algorithm to work q can be quite general, but we do need to know how to simulate from $q(\theta, \cdot)$ for every θ. ①.
2. The algorithm only requires that we know π up to constant. ②.

optimisation method vs MCMC

Choosing the proposal distribution q

optimization $l(\theta)$ find θ^* to maximise $l(\theta)$



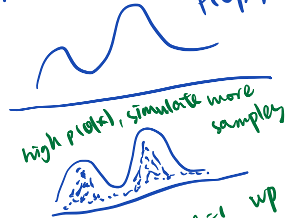
Ideally you want q to have two properties

- It proposes large moves. → efficiently walk around θ
- It leads to high average acceptance probabilities, A.

if lower average A, will simulate the same value again & again

if move is small, takes longer to visit another "hill"

MCMC



In general these two properties are competing: it is difficult to achieve both. As a result, one generally tries to choose q so that it gives "intermediate" average values for A. [An exception to this is Gibbs sampling, where $A = 1$.] around 0.3

$\theta^{(1)} = 1 \begin{cases} \theta = 1 & \text{wp } 0.9 \\ \theta' = 5 & \text{wp } 0.1 \end{cases}$ $A = 0.1$

$\theta^{(1)} = 1 \begin{cases} \theta = 1 & \text{wp. } 0.4 \\ \theta' = 10 & \text{wp } 0.6 \end{cases}$ $A = 0.6$

Random Walk Metropolis-Hastings

Perhaps the most common type of MH sampler is the "random walk" MH sampler, where the proposal distribution q involves adding a symmetric random number (e.g. $U[-\epsilon, \epsilon]$ or $N(0, \epsilon^2)$) to the current value of θ.

In this case, the terms involving q in the acceptance probability cancel, due to symmetry. That is $q(\theta, \theta') = q(\theta', \theta)$ and

$q(\theta, \theta')$

$$A = \min\left(1, \frac{\pi(\theta')}{\pi(\theta)}\right).$$

eg. $\theta' = \theta + a \to a \sim N(0, \epsilon^2)$

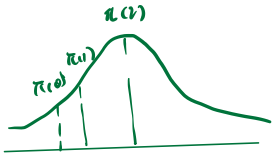
$\theta = \theta' + b \to b \sim N(0, \epsilon^2)$

so $\theta' - \theta \sim N(0, \epsilon^2)$

$\theta - \theta' \sim N(0, \epsilon^2)$

The value of ϵ determines the typical size of the proposed move, and hence the typical value for A.

Random Walk Metropolis-Hastings



The simpler form of A in the random-walk M-H makes it easier to see what the algorithm does.

$\theta^{(0)} = 1$ $\quad \left\{ \begin{array}{l} \theta = 1 \\ \theta' = 2 \end{array} \right.$ $\quad A = \min\left\{1, \frac{\pi(2)}{\pi(1)}\right\} = 1$

The proposal process has no preferences regards which part of the parameter space it wanders in.

$\pi(\theta') > \pi(\theta)$.

If the proposal process wants to move somewhere where π is larger, then $A = 1$ and the move is always accepted.

$\theta^{(1)} = 2$ $\quad \left\{ \begin{array}{l} \theta = 2 \\ \theta' = 0 \end{array} \right.$ $\quad A = \min\left\{1, \frac{\pi(0)}{\pi(2)}\right\} = \frac{\pi(0)}{\pi(2)} < 1$

$\pi(\theta') < \pi(\theta)$

However if the proposal process wants to move somewhere where π is smaller, then $A < 1$ and there is a chance we stay where we are.

In this way the M-H process will spend relatively more time in regions where π is large.

Metropolis–Hastings Algorithm

$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim p(\theta|x)$

$\theta \longrightarrow \theta'$

proposal distribution N.

Example: See `MHexample.pdf`.

$\theta' = \begin{pmatrix} \theta_1' \\ \theta_2' \end{pmatrix} \sim N\left(\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \right)$

Interactive visualization: http://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana

$\Rightarrow \theta_1' \sim N(\theta_1, \sigma^2)$

$\theta_2' \sim N(\theta_2, \sigma^2)$

in practice, use 2 types of propose together

one propose large moves

one propose smaller moves but large acceptance prob.

(problem: for multimodal, never go to other mode).

Let's go back to the Gibbs sampler

Suppose that $\theta = (\theta_1, \ldots, \theta_k)$ is a vector of unknown parameters. We aim to simulate samples from the posterior $p(\theta_1, \ldots, \theta_k | x)$. Then consider the following Markov chain

1. Start with an initial value $\theta^{(0)}$; set $t = 0$.
2. Sample $\theta_1^{(t+1)}$ from $p(\theta_1 | x, \theta_2^{(t)}, \ldots, \theta_k^{(t)})$.
3. Sample $\theta_2^{(t+1)}$ from $p(\theta_2 | x, \theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)})$.
4. $\ldots$
5. Sample $\theta_k^{(t+1)}$ from $p(\theta_k | x, \theta_1^{(t+1)}, \ldots, \theta_{k-1}^{(t+1)})$.
6. Increase t and return to 2.

Note that at each step a component of the unknown parameter is sampled from its full conditional distribution, given the data, and the current value of all other components of the parameter.

The Gibbs sampler

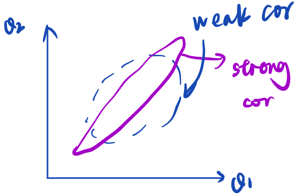
eg $\theta_1^{(t+1)}$ from $p(\theta_1 \mid x, \theta_2^{(t)}, \ldots \theta_k^{(t)})$

Indeed, this Markov chain is a special case of MH sampling: using the full conditional distributions as the proposal distribution in an MH sampler gives acceptance probability $A = 1$.

Effect of correlation on the Gibbs sampler

MH with a more efficient proposal can be efficient than Gibbs sampler

eg. $\theta = (\theta_1, \theta_2) \sim P(\theta_1, \theta_2 \mid data)$



The Gibbs sampler updates one co-ordinate at a time in order to obtain a new sample point.

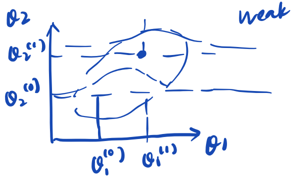
Strong correlation between the elements of $\theta = (\theta_1, \ldots, \theta_k)$ (that is, correlation between the co-ordinates of the target distribution π) will slow down the Gibbs sampler, so that it explores the sample space more slowly.

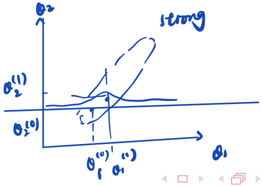
$\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)})$

We illustrate this using the Gibbs sampler for a bivariate normal. See problems from the lab later.

$\theta^{(1)} = (\theta_1^{(1)},$

simulate $\theta_1^{(1)}$ from $p(\theta_1 \mid data, \theta_2)$

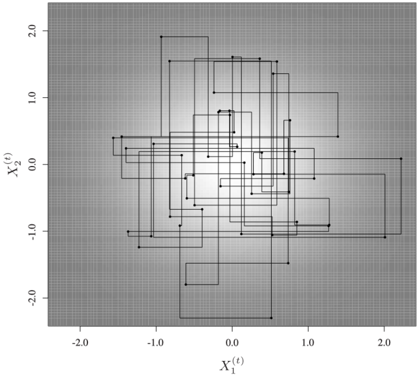




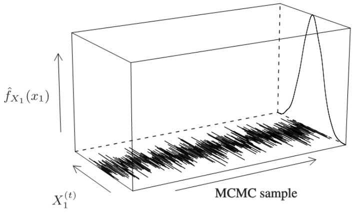
take longer to move from one end to another

Effect of correlation on the Gibbs sampler

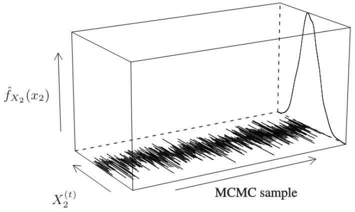
Figures from Ioana A. Cosma; Gibbs sampler aims to simulate (X_1, X_2) from bivariate standard normal distribution.



(a) First 50 iterations of $(X_1^{(t)}, X_2^{(t)})$



(b) Path of $X_1^{(t)}$ and estimated density of X after 1,000 iterations

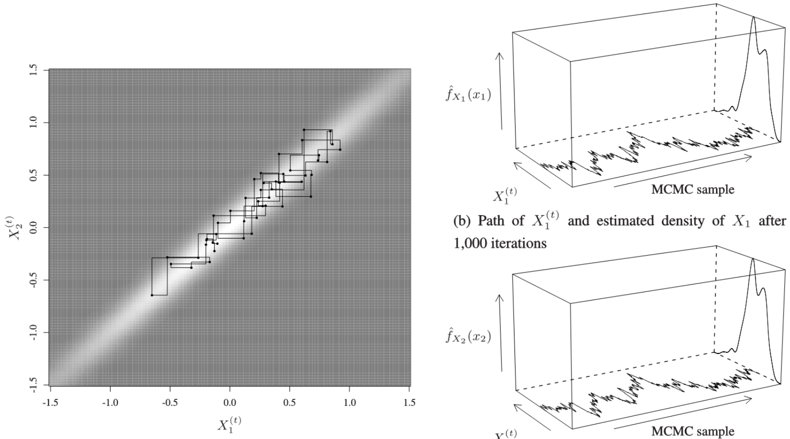


(c) Path of $X_2^{(t)}$ and estimated density of X_2 after 1,000 iterations

Figure 4.4. Gibbs sampler for a bivariate standard normal distribution with correlation $\rho(X_1, X_2) = 0.3$.

Effect of correlation on the Gibbs sampler

Figures from Ioana A. Cosma; Gibbs sampler aims to simulate (X_1, X_2) from bivariate standard normal distribution.



(a) First 50 iterations of $(X_1^{(t)}, X_2^{(t)})$

(b) Path of $X_1^{(t)}$ and estimated density of X_1 after 1,000 iterations

(c) Path of $X_2^{(t)}$ and estimated density of X_2 after 1,000 iterations

Figure 4.5. Gibbs sampler for a bivariate normal distribution with correlation $\rho(X_1, X_2) = 0.99$.

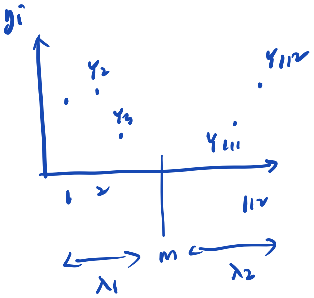
The Gibbs sampler

Interactive visualization: `http://chi-feng.github.io/mcmc-demo/app.html?algorithm=GibbsSampling&target=banana`

The Gibbs sampler: example

y_i $i=1, \cdots 112.$

1851 1962

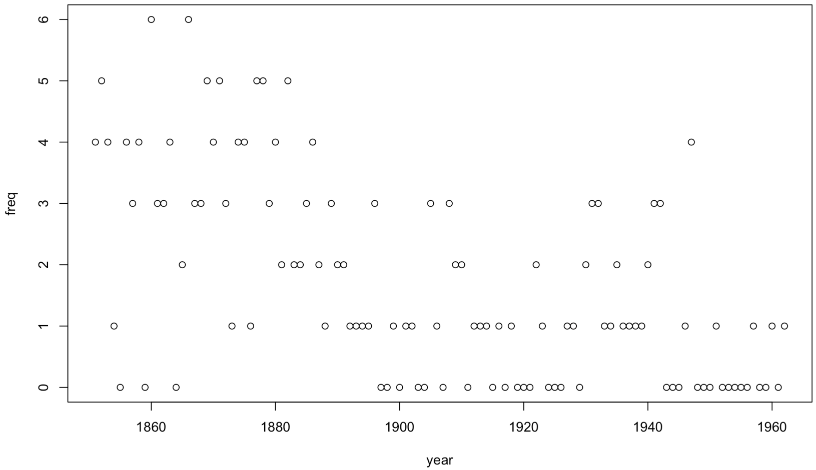


Poisson change point model

Data from Jarret (1979), A note on the intervals between coal mining disasters. *Biometrika* 66, pp. 191–193.

The data gives the dates of explosions killing 10 or more miners, from 1851 to 1962. Let Y_i be the number of such disasters in year i

```
> library(boot)
> data(coal)
> when <- floor(coal)
> year <- 1851:1962
> freq <- sapply(year, function(x, y) sum(y==x), y=when)
> n <- length(freq)
> plot(year, freq)
```


freq
0
1
2
3
4
5
6
1860
1880
1900
1920
1940
1960
year

$p(y,\lambda_1,\lambda_2,m)$

$= p(y|\lambda_1,\lambda_2,m)\cdot p(\lambda_1)\cdot p(\lambda_2)\, p(m)$

$\propto \prod_{i=1}^{m} p(y_i|\lambda_1) \prod_{i=m+1}^{n} p(y_i|\lambda_2)\cdot p(\lambda_1)\, p(\lambda_2)\, p(m)$

$\propto \prod_{i=1}^{m} \frac{e^{-\lambda_1}\lambda_1^{y_i}}{y_i!} \prod_{i=m+1}^{n} \frac{e^{-\lambda_2}\lambda_2^{y_i}}{y_i!} \cdot \lambda_1^{\alpha-1} e^{-\beta_1\lambda_1}\, \lambda_2^{\alpha_2-1} e^{-\beta_2\lambda_2}$

$f(\lambda_1|y,\lambda_2,m) \propto f(y,\lambda_1,\lambda_2,m) \propto e^{-m\lambda_1}\lambda_1^{\sum_{i=1}^m y_i}\cdot\lambda_1^{\alpha-1} e^{-\beta_1\lambda_1}$

$\propto e^{-(m+\beta_1)\lambda_1}\lambda_1^{\alpha+\sum y_i-1}$

$\sim \Gamma(\alpha+\sum_{i=1}^m y_i,\ m+\beta_1)$

Was there a change in the rate of disasters? To (help) answer this question we use the following model:

$$
\begin{aligned}
Y_i &\sim \text{pois}(\lambda_1), \text{ for } i = 1, \ldots, m \\
Y_i &\sim \text{pois}(\lambda_2), \text{ for } i = m+1, \ldots, n.
\end{aligned}
$$

To find m

We can impose the following priors on λ_1, λ_2, m:

$$
\begin{aligned}
\lambda_1 &\sim \Gamma(\alpha_1, \beta_1) \\
\lambda_2 &\sim \Gamma(\alpha_2, \beta_2) \\
m &\sim U\{1, \ldots, n\}.
\end{aligned}
$$

$X \sim \Gamma(\alpha, \beta)$

$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$

want to get

$p(\lambda_1 | \lambda_2, m, y) \propto p(\lambda_1, \lambda_2, m, y)$

$f(\lambda_2|y,\lambda_1,m) \sim \Gamma(\alpha_2+\sum_{i=m+1}^n y_i,\ \beta_2+n-m)$

$p(\lambda_2 | \lambda_1, m, y) \propto p(\lambda_1, \lambda_2, m, y)$

$p(m | \lambda_1, \lambda_2, y) \propto p(\lambda_1, \lambda_2, m, y)$

$f(m|\lambda_1,\lambda_2,y) \propto e^{-\lambda_1 m}\lambda_1^{\sum_{i=1}^m y_i} e^{-(n-m)\lambda_2}\lambda_2^{\sum_{i=m+1}^n y_i}$

$\propto e^{-m(\lambda_1-\lambda_2)}\lambda_1^{\sum_{i=1}^m y_i}\cdot(\lambda_2^{-\sum_{i=1}^m y_i})$

$\propto e^{-m(\lambda_1-\lambda_2)}\left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{i=1}^m y_i}$

Then, the joint density is given by

$$
p(y, \lambda_1, \lambda_2, m) \propto \prod_{i=1}^{m} \frac{e^{-\lambda_1}\lambda_1^{y_i}}{y_i!} \prod_{j=m+1}^{n} \frac{e^{-\lambda_2}\lambda_2^{y_j}}{y_j!} \lambda_1^{\alpha_1-1} e^{-\beta_1\lambda_1} \lambda_2^{\alpha_2-1} e^{-\beta_2\lambda_2}.
$$

$p(\lambda_1,\lambda_2,m,y) = p(y|\lambda_1,\lambda_2,m)\cdot p(\lambda_1,\lambda_2,m)$

$= \prod_{i=1}^m p(y|\lambda_1) \prod_{i=m+1}^n p(y|\lambda_2)\, \underbrace{p(\lambda_1)\cdot p(\lambda_2)\cdot p(m)}_{\text{independent prior}}$

$\propto \prod_{i=1}^{m} e^{-\lambda_1} \lambda_1^{y_i} \prod_{i=m+1}^{n} e^{-\lambda_2} \lambda_2^{y_i} \lambda_1^{\alpha_1-1} e^{-\beta_1\lambda_1} \lambda_2^{\alpha_2-1} e^{-\beta_2\lambda_2}$

Collecting like terms we get the posterior

$= e^{-m\lambda_1} \lambda_1^{\sum_{i=1}^{m} y_i} e^{-(n-m)\lambda_2} \lambda_2^{\sum_{i=m+1}^{n} y_i} \lambda_1^{\alpha_1-1} e^{-\beta_1\lambda_1} \lambda_2^{\alpha_2-1} e^{-\beta_2\lambda_2}$

$$
\begin{aligned}
p(\lambda_1, \lambda_2, m|\text{y}) \quad \propto \quad & e^{-(m+\beta_1)\lambda_1} \lambda_1^{\sum_{i=1}^{m} y_i + \alpha_1 - 1} \\
& \times e^{-(n-m+\beta_2)\lambda_2} \lambda_2^{\sum_{j=m+1}^{n} y_j + \alpha_2 - 1}
\end{aligned}
$$

$p(\lambda_1|\lambda_2, m, y) \propto p(\lambda_1, \lambda_2, m, y)$

$= e^{-(m+\beta_1)\lambda_1} \lambda_1^{\sum_{i=1}^{m} y_i + \alpha_1 - 1}$

$\sim \Gamma(\sum_{i=1}^{m} y_i + \alpha_1, \beta_1 + m)$

and thus the conditioned distributions are

similarly $p(\lambda_2|\lambda_1, m, y)$

$\sim \Gamma(\sum_{i=m+1}^{n} y_i + \alpha_2, \beta_2 + n - m)$

$$
\begin{aligned}
p(\lambda_1|\lambda_2, m, \text{y}) \quad & \sim \quad \Gamma(\alpha_1 + \sum_{i=1}^{m} y_i, \beta_1 + m) \\
p(\lambda_2|\lambda_1, m, \text{y}) \quad & \sim \quad \Gamma(\alpha_2 + \sum_{j=m+1}^{n} y_j, \beta_2 + n - m) \\
p(m|\lambda_1, \lambda_2, \text{y}) \quad & \propto \quad \lambda_1^{\sum_{i=1}^{m} y_i} \lambda_2^{\sum_{j=m+1}^{n} y_j} e^{(\lambda_2 - \lambda_1)m} \\
& \propto \quad \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{i=1}^{m} y_i} e^{(\lambda_2 - \lambda_1)m}.
\end{aligned}
$$

eg. $m = 1, 2, 3$ given λ_1, λ_2, y known

$p(m=1|\lambda_1, \lambda_2, y) \propto 10 \rightarrow$ wp. $\frac{10}{10+11+19}$

$p(m=2|\lambda_1, \lambda_2, y) \propto 11$ wp $\frac{11}{10+11+19}$

$p(m=3|\lambda_1, \lambda_2, y) \propto 19$ wp. $\frac{19}{10+11+19}$

$p(m|\lambda_1, \lambda_2, y)$

$\propto p(\lambda_1, \lambda_2, m, y)$ $\quad \sum_{i=1}^{n} y_i - \sum_{i=1}^{m} y_i$

$\propto e^{-m\lambda_1} \lambda_1^{\sum_{i=1}^{m} y_i} e^{-(n-m)\lambda_2} \lambda_2^{\sum_{i=m+1}^{n} y_i}$

$\propto e^{m(\lambda_2 - \lambda_1)} (\frac{\lambda_1}{\lambda_2})^{\sum_{i=1}^{m} y_i}$

Note that $p(m|\lambda_1, \lambda_2, y)$ is not a known distribution, but it is finite so we can easily simulate samples from this distribution.

We impliment a Gibbs sampler in R for this model.

See `Gibbs_example_coal.pdf`

Learning goals

Metropolis-Hastings Algorithm:

- Understand key ideas for the algorithm
- Be able to derive and implement the algorithm
- (not examinable; challenging) Obtain intuitions on the theory to support why the algorithm works

Gibbs sampler:

- Understand it is a special case of the Metropolis-Hastings Algorithm
- Understand its potential limitation compared to the Metropolis-Hastings Algorithm