

## Module 6: Introduction to Gibbs Sampling

Rebecca C. Steorts

# Agenda

- ▶ Gibbs sampling
- ▶ Exponential example
- ▶ Normal example
- ▶ Pareto example

# Gibbs sampler

- ▶ Suppose  $p(x, y)$  is a p.d.f. or p.m.f. that is difficult to sample from directly.
- ▶ Suppose, though, that we *can* easily sample from the conditional distributions  $p(x|y)$  and  $p(y|x)$ .
- ▶ The Gibbs sampler proceeds as follows:
  1. set  $x$  and  $y$  to some initial starting values
  2. then sample  $x|y$ , then sample  $y|x$ , then  $x|y$ , and so on.

# Gibbs sampler

0. Set  $(x_0, y_0)$  to some starting value.
1. Sample  $x_1 \sim p(x|y_0)$ , that is, from the conditional distribution  $X \mid Y = y_0$ .  
Current state:  $(x_1, y_0)$   
Sample  $y_1 \sim p(y|x_1)$ , that is, from the conditional distribution  $Y \mid X = x_1$ .  
Current state:  $(x_1, y_1)$
2. Sample  $x_2 \sim p(x|y_1)$ , that is, from the conditional distribution  $X \mid Y = y_1$ .  
Current state:  $(x_2, y_1)$   
Sample  $y_2 \sim p(y|x_2)$ , that is, from the conditional distribution  $Y \mid X = x_2$ .  
Current state:  $(x_2, y_2)$
- $\vdots$

Repeat iterations 1 and 2, M times.

## Gibbs sampler

This procedure defines a sequence of pairs of random variables

$$(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots$$

# Markov chain and dependence

$$(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots$$

satisfies the property of being a Markov chain.

The conditional distribution of  $(X_i, Y_i)$  given all of the previous pairs depends only on  $(X_{i-1}, Y_{i-1})$

$(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots$  are not iid samples (Think about why).

# Ideal Properties of MCMC

- ▶  $(x_0, y_0)$  chosen to be in a region of high probability under  $p(x, y)$ , but often this is not so easy.
- ▶ We run the chain for  $M$  iterations and discard the first  $B$  samples  $(X_1, Y_1), \dots, (X_B, Y_B)$ . This is called *burn-in*.
- ▶ Typically: if you run the chain long enough, the choice of  $B$  doesn't matter.
- ▶ Roughly speaking, the performance of an MCMC algorithm—that is, how quickly the sample averages  $\frac{1}{N} \sum_{i=1}^N h(X_i, Y_i)$  converge—is referred to as the *mixing rate*.
- ▶ An algorithm with good performance is said to “have good mixing”, or “mix well”.

## Exponential Example

Consider the following Exponential model for observation(s)  
 $x = (x_1, \dots, x_n)$ .<sup>1</sup>:

$$p(x|a, b) = ab \exp(-abx)I(x > 0)$$

and suppose the prior is

$$p(a, b) = \exp(-a - b)I(a, b > 0).$$

You want to sample from the posterior  $p(a, b|x)$ .

---

<sup>1</sup>Please note that in the attached data there are 40 observations, which can be found in data-exponential.csv.



## Conditional distributions

$$\begin{aligned} p(\mathbf{x}|a, b) &= \prod_{i=1}^n p(x_i|a, b) \\ &= \prod_{i=1}^n ab \exp(-abx_i) \\ &= (ab)^n \exp\left(-ab \sum_{i=1}^n x_i\right). \end{aligned}$$

The function is symmetric for  $a$  and  $b$ , so we only need to derive  $p(a|\mathbf{x}, b)$ .

# Conditional distributions

This conditional distribution satisfies

$$\begin{aligned} p(a|\mathbf{x}, b) &\propto_a p(a, b, \mathbf{x}) \\ &= p(\mathbf{x}|a, b)p(a, b) \\ &= \text{fill in full details for homework} \end{aligned}$$

## Gibbs sampling code

```
knitr::opts_chunk$set(cache=TRUE)
library(MASS)
data <- read.csv("data-exponential.csv", header = FALSE)
```

## Gibbs sampling code

```
#####  
# This function is a Gibbs sampler  
#  
# Args  
#   start.a: initial value for a  
#   start.b: initial value for b  
#   n.sims: number of iterations to run  
#   data:   observed data, should be in a  
#           # data frame with one column  
#  
# Returns:  
#   A two column matrix with samples  
#       #   for a in first column and  
#   samples for b in second column  
#####
```

## Gibbs sampling code

```
sampleGibbs <- function(start.a, start.b, n.sims, data){  
  # get sum, which is sufficient statistic  
  x <- sum(data)  
  # get n  
  n <- nrow(data)  
  # create empty matrix, allocate memory for efficiency  
  res <- matrix(NA, nrow = n.sims, ncol = 2)  
  res[1,] <- c(start.a, start.b)  
  for (i in 2:n.sims){  
    # sample the values  
    res[i,1] <- rgamma(1, shape = n+1,  
                      rate = res[i-1,2]*x+1)  
    res[i,2] <- rgamma(1, shape = n+1,  
                      rate = res[i,1]*x+1)  
  }  
  return(res)  
}
```

## Gibbs sampler code

```
# run Gibbs sampler  
n.sims <- 10000  
res <- sampleGibbs(.25,.25,n.sims,data)  
head(res)
```

```
##           [,1]      [,2]  
## [1,] 0.2500000 0.2500000  
## [2,] 1.1785823 0.4673874  
## [3,] 1.0490590 0.6020879  
## [4,] 0.7366618 0.7891401  
## [5,] 0.6151383 0.6129127  
## [6,] 0.8943432 0.5312112
```

## Toy Example

- ▶ The Gibbs sampling approach is to alternately sample from  $p(x|y)$  and  $p(y|x)$ .
- ▶ Note  $p(x, y)$  is symmetric with respect to  $x$  and  $y$ .
- ▶ Hence, only need to derive one of these and then we can get the other one by just swapping  $x$  and  $y$ .
- ▶ Let's look at  $p(x|y)$ .

## Toy Example

$$p(x, y) \propto e^{-xy} \mathbb{1}(x, y \in (0, c))$$

$$p(x|y) \underset{x}{\propto} p(x, y) \underset{x}{\propto} e^{-xy} \mathbb{1}(0 < x < c) \underset{x}{\propto} \text{Exp}(x|y) \mathbb{1}(x < c).^2$$

- ▶  $p(x|y)$  is a *truncated* version of the  $\text{Exp}(y)$  distribution
- ▶ It is the same as taking  $X \sim \text{Exp}(y)$  and conditioning on it being less than  $c$ , i.e.,  $X \mid X < c$ .
- ▶ Let's refer to this as the  $\text{TExp}(y, (0, c))$  distribution.

---

<sup>2</sup>Under  $\propto_x$ , we write the random variable ( $x$ ) for clarity.



## Toy Example

An easy way to generate a sample from  $Z \sim \text{TExp}(\theta, (0, c))$ , is:

1. Sample  $U \sim \text{Uniform}(0, F(c|\theta))$  where

$$F(x|\theta) = 1 - e^{-\theta x}$$

is the  $\text{Exp}(\theta)$  c.d.f.

2. Set  $Z = F^{-1}(U|\theta)$  where

$$F^{-1}(u|\theta) = -(1/\theta) \log(1 - u)$$

is the inverse c.d.f. for  $u \in (0, 1)$ .

Hint: To verify the last step: apply the rejection principle (along with the inverse cdf technique). Verify the last step on your own.

## Toy example

Let's apply Gibbs sampling, denoting  $S = (0, c)$ .

0. Initialize  $x_0, y_0 \in S$ .
1. Sample  $x_1 \sim \text{TExp}(y_0, S)$ , then sample  $y_1 \sim \text{TExp}(x_1, S)$ .
2. Sample  $x_2 \sim \text{TExp}(y_1, S)$ , then sample  $y_2 \sim \text{TExp}(x_2, S)$ .
- $\vdots$
- $N$ . Sample  $x_N \sim \text{TExp}(y_{N-1}, S)$ , sample  $y_N \sim \text{TExp}(x_N, S)$ .

Figure 1 demonstrates the algorithm, with  $c = 2$  and initial point  $(x_0, y_0) = (1, 1)$ .

# Toy example

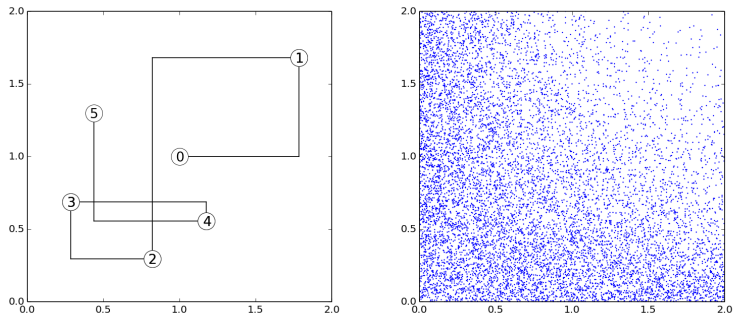


Figure 1: (Left) Schematic representation of the first 5 Gibbs sampling iterations/sweeps/scans. (Right) Scatterplot of samples from  $10^4$  Gibbs sampling iterations.

## Example: Normal with semi-conjugate prior

Consider  $X_1, \dots, X_n | \mu, \lambda \stackrel{iid}{\sim} \mathcal{N}(\mu, \lambda^{-1})$ . Then independently consider

$$\mu \sim \mathcal{N}(\mu_0, \lambda_0^{-1})$$

$$\lambda \sim \text{Gamma}(a, b)$$

This is called a semi-conjugate situation, in the sense that the prior on  $\mu$  is conjugate for each fixed value of  $\lambda$ , and the prior on  $\lambda$  is conjugate for each fixed value of  $\mu$ .

For ease of notation, denote the observed data points by  $x_{1:n}$ .

## Example

We know that for the Normal–Normal model, we know that for any fixed value of  $\lambda$ ,

$$\mu|\lambda, x_{1:n} \sim \mathcal{N}(M_\lambda, L_\lambda^{-1})$$

where

$$L_\lambda = \lambda_0 + n\lambda \quad \text{and} \quad M_\lambda = \frac{\lambda_0\mu_0 + \lambda \sum_{i=1}^n x_i}{\lambda_0 + n\lambda}.$$

For any fixed value of  $\mu$ , it is straightforward to derive<sup>3</sup> that

$$\lambda|\mu, x_{1:n} \sim \text{Gamma}(A_\mu, B_\mu) \tag{1}$$

where  $A_\mu = a + n/2$  and

$$B_\mu = b + \frac{1}{2} \sum (x_i - \mu)^2 = n\hat{\sigma}^2 + n(\bar{x} - \mu)^2$$

where  $\hat{\sigma}^2 = \frac{1}{n} \sum (x_i - \bar{x})^2$ .

<sup>3</sup>do this on your own

## Example

To implement Gibbs sampling in this example, each iteration consists of sampling:

$$\begin{aligned}\mu|\lambda, x_{1:n} &\sim \mathcal{N}(M_\lambda, L_\lambda^{-1}) \\ \lambda|\mu, x_{1:n} &\sim \text{Gamma}(A_\mu, B_\mu).\end{aligned}$$

# Pareto example

Distributions of sizes and frequencies often tend to follow a “power law” distribution.

- ▶ wealth of individuals
- ▶ size of oil reserves
- ▶ size of cities
- ▶ word frequency
- ▶ returns on stocks

# Power law distribution

The Pareto distribution with shape  $\alpha > 0$  and scale  $c > 0$  has p.d.f.

$$\text{Pareto}(x|\alpha, c) = \frac{\alpha c^\alpha}{x^{\alpha+1}} \mathbb{1}(x > c) \propto \frac{1}{x^{\alpha+1}} \mathbb{1}(x > c).$$

- ▶ This is referred to as a power law distribution, because the p.d.f. is proportional to  $x$  raised to a power.
- ▶  $c$  is a lower bound on the observed values.
- ▶ We will use Gibbs sampling to perform inference for  $\alpha$  and  $c$ .



## Pareto example

Rank	City	Population
1	Charlotte	731424
2	Raleigh	403892
3	Greensboro	269666
4	Durham	228330
5	Winston-Salem	229618
6	Fayetteville	200564
7	Cary	135234
8	Wilmington	106476
9	High Point	104371
10	Greenville	84554
11	Asheville	85712
12	Concord	79066
⋮	⋮	⋮
44	Havelock	20735
45	Carrboro	19582
46	Shelby	20323
47	Clemmons	18627
48	Lexington	18931
49	Elizabeth City	18683

# Parameter interpretations

- ▶  $\alpha$  tells us the scaling relationship between the size of cities and their probability of occurring.
  - ▶ Let  $\alpha = 1$ .
  - ▶ Density looks like  $1/x^{\alpha+1} = 1/x^2$ .
  - ▶ Cities with 10,000–20,000 inhabitants occur roughly  $10^{\alpha+1} = 100$  times as frequently as cities with 100,000–110,000 inhabitants.
- ▶  $c$  represents the cutoff point—any cities smaller than this were not included in the dataset.

## Prior selection

For simplicity, let's use an **(improper) default prior**:

$$p(\alpha, c) \propto \mathbb{1}(\alpha, c > 0).$$

Recall:

- ▶ An *improper/default prior* is a non-negative function of the parameters which integrates to infinity.
- ▶ Often (but not always!) the resulting “posterior” will be proper.
- ▶ It is important that the “posterior” be proper, since otherwise the whole Bayesian framework breaks down.

## Pareto example

Recall

$$p(x|\alpha, c) = \frac{\alpha c^\alpha}{x^{\alpha+1}} \mathbb{1}(x > c) \quad (2)$$

$$\mathbb{1}(\alpha, c > 0) \quad (3)$$

Let's derive the posterior:

$$\begin{aligned} p(\alpha, c|x_{1:n}) &\stackrel{\text{def}}{\propto}_{\alpha, c} p(x_{1:n}|\alpha, c)p(\alpha, c) \\ &\propto_{\alpha, c} \mathbb{1}(\alpha, c > 0) \prod_{i=1}^n \frac{\alpha c^\alpha}{x_i^{\alpha+1}} \mathbb{1}(x_i > c) \\ &= \frac{\alpha^n c^{n\alpha}}{(\prod x_i)^{\alpha+1}} \mathbb{1}(c < x_*) \mathbb{1}(\alpha, c > 0) \end{aligned} \quad (4)$$

where  $x_* = \min\{x_1, \dots, x_n\}$ .

## Pareto example

As a joint distribution on  $(\alpha, c)$ ,

- ▶ this does not seem to have a recognizable form,
- ▶ and it is not clear how we might sample from it directly.

## Gibbs sampling

Let's try Gibbs sampling! To use Gibbs, we need to be able to sample  $\alpha|c, x_{1:n}$  and  $c|\alpha, x_{1:n}$ .

By Equation 4, we find that

$$\begin{aligned} p(\alpha|c, x_{1:n}) &\propto_{\alpha} p(\alpha, c|x_{1:n}) \propto_{\alpha} \frac{\alpha^n c^{n\alpha}}{(\prod x_i)^{\alpha}} \mathbb{1}(\alpha > 0) \\ &= \alpha^n \exp(-\alpha(\sum \log x_i - n \log c)) \mathbb{1}(\alpha > 0) \\ &\propto_{\alpha} \text{Gamma}(\alpha | n + 1, \sum \log x_i - n \log c), \end{aligned}$$

and

$$p(c|\alpha, x_{1:n}) \propto_c p(\alpha, c|x_{1:n}) \propto_c c^{n\alpha} \mathbb{1}(0 < c < x_*),$$

which we will define to be  $\text{Mono}(\alpha, x_*)$

## Mono distribution

For  $a > 0$  and  $b > 0$ , define the distribution  $\text{Mono}(a, b)$  (for monomial) with p.d.f.

$$\text{Mono}(x|a, b) \propto x^{a-1} \mathbb{1}(0 < x < b).$$

Since  $\int_0^b x^{a-1} dx = b^a/a$ , we have

$$\text{Mono}(x|a, b) = \frac{a}{b^a} x^{a-1} \mathbb{1}(0 < x < b),$$

and for  $0 < x < b$ , the c.d.f. is

$$F(x|a, b) = \int_0^x \text{Mono}(y|a, b) dy = \frac{a}{b^a} \frac{x^a}{a} = \frac{x^a}{b^a}.$$

## Pareto example

To use the inverse c.d.f. technique, we solve for the inverse of  $F$  on  $0 < x < b$ : Let  $u = \frac{x^a}{b^a}$  and solve for  $x$ .

$$u = \frac{x^a}{b^a} \tag{5}$$

$$b^a u = x^a \tag{6}$$

$$bu^{1/a} = x \tag{7}$$

Can sample from  $\text{Mono}(a, b)$  by drawing  $U \sim \text{Uniform}(0, 1)$  and setting  $X = bU^{1/a}$ .<sup>4</sup>

---

<sup>4</sup>It turns out that this is an inverse of the Pareto distribution, in the sense that if  $X \sim \text{Pareto}(\alpha, c)$  then  $1/X \sim \text{Mono}(\alpha, 1/c)$ .



## Pareto example

So, in order to use the Gibbs sampling algorithm to sample from the posterior  $p(\alpha, c | x_{1:n})$ , we initialize  $\alpha$  and  $c$ , and then alternately update them by sampling:

$$\begin{aligned}\alpha | c, x_{1:n} &\sim \text{Gamma}(n + 1, \sum \log x_i - n \log c) \\ c | \alpha, x_{1:n} &\sim \text{Mono}(n\alpha + 1, x_*).\end{aligned}$$

# Traceplots

**Traceplots.** A traceplot simply shows the sequence of samples, for instance  $\alpha_1, \dots, \alpha_N$ , or  $c_1, \dots, c_N$ . Traceplots are a simple but very useful way to visualize how the sampler is behaving.

# Traceplots

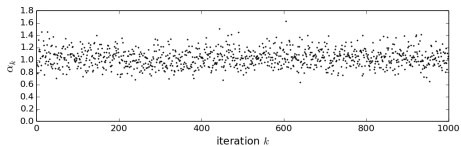


Figure 2: Traceplot of  $\alpha$

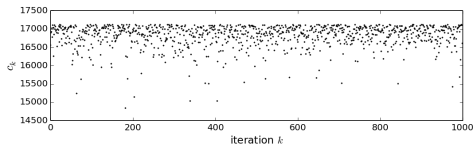


Figure 3: Traceplot of  $c$ .

## Estimated density

**Estimated density.** We are primarily interested in the posterior on  $\alpha$ , since it tells us the scaling relationship between the size of cities and their probability of occurring.

By making a histogram of the samples  $\alpha_1, \dots, \alpha_N$ , we can estimate the posterior density  $p(\alpha|x_{1:n})$ .

The two vertical lines indicate the lower  $\ell$  and upper  $u$  boundaries of an (approximate) 90% credible interval  $[\ell, u]$ —that is, an interval that contains 90% of the posterior probability:

$$\mathbb{P}(\alpha \in [\ell, u] | x_{1:n}) = 0.9.$$

## Estimated density

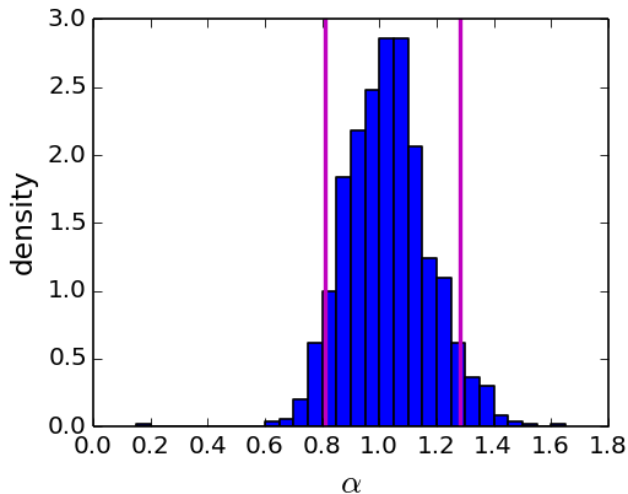


Figure 4: Estimated density of  $\alpha|x_{1:n}$  with  $\approx 90$  percent credible intervals.

## Running averages

**Running averages.** Panel (d) shows the running average  $\frac{1}{k} \sum_{i=1}^k \alpha_i$  for  $k = 1, \dots, N$ .

In addition to traceplots, running averages such as this are a useful heuristic for visually assessing the convergence of the Markov chain.

The running average shown in this example still seems to be meandering about a bit, suggesting that the sampler needs to be run longer (but this would depend on the level of accuracy desired).

## Running averages

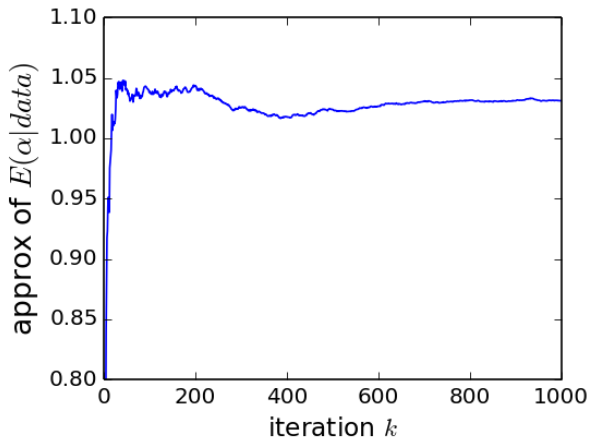


Figure 5: Running average plot

# Survival functions

A survival function is defined to be

$$S(x) = \mathbb{P}(X > x) = 1 - \mathbb{P}(X \leq x).$$

Power law distributions are often displayed by plotting their survival function  $S(x)$ , on a log-log plot.

Why?  $S(x) = (c/x)^\alpha$  for the  $\text{Pareto}(\alpha, c)$  distribution and on a log-log plot this appears as a line with slope  $-\alpha$ .

The posterior survival function (or more precisely, the posterior predictive survival function), is  $S(x|x_{1:n}) = \mathbb{P}(X_{n+1} > x \mid x_{1:n})$ .



## Survival functions

Figure 6(e) shows an empirical estimate of the survival function (based on the empirical c.d.f.,  $\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(x \geq x_i)$ ) along with the posterior survival function, approximated by

$$S(x|x_{1:n}) = \mathbb{P}(X_{n+1} > x \mid x_{1:n}) \quad (8)$$

$$= \int \mathbb{P}(X_{n+1} > x \mid \alpha, c) p(\alpha, c | x_{1:n}) d\alpha dc \quad (9)$$

$$\approx \frac{1}{N} \sum_{i=1}^N \mathbb{P}(X_{n+1} > x \mid \alpha_i, c_i) = \frac{1}{N} \sum_{i=1}^N (c_i/x)^{\alpha_i}. \quad (10)$$

This is computed for each  $x$  in a grid of values.

# Survival functions

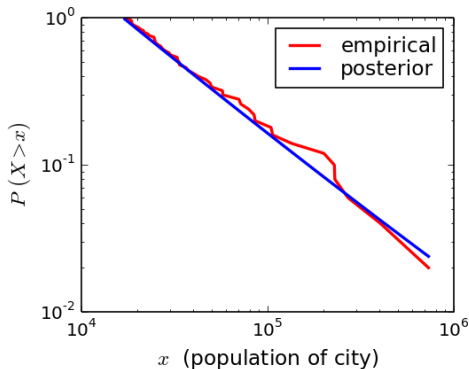


Figure 6: Empirical vs posterior survival function

How could we get a better empirical approximation?