

Bayesian Hierarchical Modeling for the Social Sciences

Markov Chain Monte Carlo Methods

JEFF GILL

Department of Government

Department of Mathematics and Statistics

Center for Behavioral Neurosciences

American University

Status

- ▶ This is the official start of the second half of the Workshop.

- ▶ Our methodological roadmap:

<i>tool</i>	<i>capabilities</i>	<i>limitations</i>
using lmer in R	basic models, good understanding	normally distributed group levels, limited link functions
jags and bugs	builds on R , multiple hierarchies, no distributional restrictions intuitive modeling language	none

- ▶ **jags** and **bugs** are completely free implementations of *Markov chain Monte Carlo*, which revolutionized statistics starting in 1990.

Bureaucratic Politics Example

- ▶ Contains *every* federal political appointee to full-time positions requiring Senate confirmation from November, 1964 through December, 1984 (collected by Mackenzie and Light, ICPSR Study Number 8458, Spring 1987).
- ▶ The survey queries various aspects of the Senate confirmation process, acclimation to running an agency or program, and relationships with other functions of government.
- ▶ The authors needed to preserve anonymity so they embargoed some variables and randomly sampled 1,500 down to 512.
- ▶ These latter issues are dealt with in Gill and Casella (JASA 2009).

Bureaucratic Politics Example

- ▶ **Outcome Variable:** **stress** as a surrogate measure for self-perceived effectiveness and job-satisfaction, measured as a five-point scale from “not stressful at all” to “very stressful.”

- ▶ **Explanatory Variables:**
 - ▶ Government Experience,
 - ▶ Ideology,
 - ▶ Committee Relationship,
 - ▶ Career.Exec-Compet,
 - ▶ Career.Exec-Liaison/Bur,
 - ▶ Career.Exec-Liaison/Cong,
 - ▶ Career.Exec-Day2day,
 - ▶ Career.Exec-Diff,
 - ▶ Confirmation Preparation,
 - ▶ Hours/Week,
 - ▶ President Orientation.

Ordered Logit Model

- ▶ A Bayesian random effects specification for ordered survey outcomes, so latent variable thresholds for \mathbf{Y} are assumed on the ordering:

$$\mathbf{U}_i : \theta_0 \begin{array}{c} \longleftarrow \\ \xrightarrow{c=1} \end{array} \theta_1 \begin{array}{c} \longleftarrow \\ \xrightarrow{c=2} \end{array} \theta_2 \begin{array}{c} \longleftarrow \\ \xrightarrow{c=3} \end{array} \theta_3 \dots \theta_{C-1} \begin{array}{c} \longleftarrow \\ \xrightarrow{c=C} \end{array} \theta_C$$

- ▶ The vector of (unseen) utilities across individuals in the sample, \mathbf{U} , is determined by a linear additive specification of explanatory variables: $\mathbf{U} = -\mathbf{X}'\boldsymbol{\gamma} + \boldsymbol{\eta}$, where $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_p]$ does not depend on the θ_j , and $\boldsymbol{\eta} \sim F_{\boldsymbol{\eta}}$.
- ▶ This means that the probability that individual i in the sample is observed to be in category r or lower is:

$$P(\mathbf{Y}_i \leq r | \mathbf{X}_i) = P(\mathbf{U}_i \leq \theta_r) = P(\boldsymbol{\eta} \leq \theta_r + \mathbf{X}'_i \boldsymbol{\gamma}) = F_{\boldsymbol{\eta}_i}(\theta_r + \mathbf{X}'_i \boldsymbol{\gamma}),$$

which is differently signed than in \mathbf{R} : “*logit* $P(Y \leq k|x) = \text{zeta}.k - \text{eta}$ ” from the help page.

- ▶ Specifying a logistic distributional assumption on the errors and adding the random effect term produces this logistic cumulative specification for the whole sample:

$$F_{\boldsymbol{\eta}}(\theta_r + \mathbf{X}'\boldsymbol{\gamma} + \mathbf{b}) = P(\mathbf{Y} \leq r | \mathbf{X}) = [1 + \exp(-\theta_r - \mathbf{X}'\boldsymbol{\gamma} + \mathbf{b})]^{-1}$$

Model Priors

- ▶ The software we will use is for Bayesian hierarchical models, meaning that we have to stipulate an assumed distribution for the unknown parameters *before* observing and analyzing the data.
- ▶ Most social scientists give fairly vague prior statements as a way not to worry about this.
- ▶ Prior distributions are either semi-informed or skeptical:

$p(\boldsymbol{\gamma}_k) \sim \mathcal{N}(\mu_{\boldsymbol{\gamma}_k}, \sigma_{\boldsymbol{\gamma}}^2)$, $k = 1, \dots, p$ for each of the p explanatory variables,

$p(\boldsymbol{\theta}_j) \sim \mathcal{N}(0, \sigma_{\boldsymbol{\theta}}^2)$, $j = 1, \dots, C - 1$ for the four latent variable thresholds,

$b_i \sim \mathcal{N}(0, \tau)$ for the random effects term,

$\tau \sim \mathcal{IG}(\delta_1, \delta_2)$ for the random effects hyperprior,

Model Posterior

- ▶ All this produces a posterior distribution according to:

$$\begin{aligned} \pi(\boldsymbol{\gamma}, \boldsymbol{\theta} | \mathbf{X}, \mathbf{Y}) &\propto L(\boldsymbol{\gamma}, \boldsymbol{\theta} | \mathbf{X}, \mathbf{Y}) p(\boldsymbol{\theta}) p(\boldsymbol{\gamma}) p(b | \tau) p(\tau) \\ &\propto \prod_{i=1}^n \prod_{j=1}^{C-1} \prod_{k=1}^p [\Lambda(\theta_j - \mathbf{X}'_i \boldsymbol{\gamma} + \mathbf{b}_i) - \Lambda(\theta_{j-1} - \mathbf{X}'_i \boldsymbol{\gamma} + \mathbf{b}_i)]^{z_{ij}} \\ &\quad \times \exp \left(-\frac{(\boldsymbol{\gamma}_k - \mu_{\boldsymbol{\gamma}_k})^2}{2\sigma_{\boldsymbol{\gamma}}^2} - \frac{\theta_j^2}{2\sigma_{\theta}^2} - \frac{b_i^2}{2\tau^2} - \frac{\delta_2}{\tau} \right) \tau^{-(\delta_1+1)} \end{aligned}$$

which is kind of ugly (and hard marginalize).

- ▶ While this form tells us everything we need to know about the *joint distribution* of the model parameters, we need to marginalize (integrate) it for every one of these parameters to create an informative regression table.
- ▶ Solution: Gibbs sampling (a type of MCMC), which is a special application of a special kind of stochastic process..

Posterior Summary, Model for Survey of Political Executives

	Mean	Std.Err.	95% HPD Intervals
<i>Explanatory Variables:</i>			
Constant Term	1.20215	2.24169	
Government Experience	-0.65500	0.60664	
Ideology	-0.49140	0.32964	
Committee Relationship	1.14550	0.39131	
Career.Exec-Compet	1.73300	0.85088	
Career.Exec-Liaison/Bur	-2.81800	0.55620	
Career.Exec-Liaison/Cong	1.03675	0.49193	
Career.Exec-Day2day	-0.76595	0.38019	
Career.Exec-Diff	0.27780	0.29838	
Confirmation Preparation	1.02440	0.45753	
Hours/Week	-0.72720	0.42732	
President Orientation	1.94950	0.86787	
<i>Threshold Intercepts:</i>			
None Little	-5.93500	1.85782	
Little Some	-2.69250	1.59126	
Some Significant	1.19300	1.52653	
Significant Extreme	8.40450	2.10379	
$\hat{\sigma} = 6.04350$ (1.20325), dashed vertical line at zero.			

What is a Stochastic Process?

- ◇ A type of stochastic process that will help us estimate posterior quantities.
- ◇ A *stochastic process* is a consecutive set of random quantities defined on some known state space, Θ , indexed so that the order is known: $\{\theta^{[t]}: t \in T\}$.
- ◇ Frequently, but not necessarily, T is the set of positive integers implying consecutive, even-spaced time intervals: $\{\theta^{[t=0]}, \theta^{[t=1]}, \theta^{[t=2]}, \dots\}$.
- ◇ A stochastic process must also be defined with respect to a *state space*, Θ , which identifies the range of possible values of θ . This state space is either discrete or continuous depending on how the variable of interest is measured.

What is a Markov Chain?

◇ A *Markov chain* is a stochastic process with the property that any specified state in the series, $\theta^{[t]}$, is dependent only the previous value of the chain, $\theta^{[t-1]}$.

◇ Therefore values are *conditionally* independent of all other previous values: $\theta^{[0]}, \theta^{[1]}, \dots, \theta^{[t-2]}$.

◇ Formally:

$$P(\theta^{[t]} \in A | \theta^{[0]}, \theta^{[1]}, \dots, \theta^{[t-2]}, \theta^{[t-1]}) = P(\theta^{[t]} \in A | \theta^{[t-1]}),$$

where A is any identified set (an event or range of events) on the complete state space. (We will use this A notation extensively.)

◇ Colloquially:

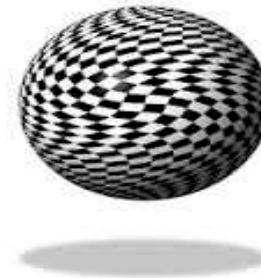
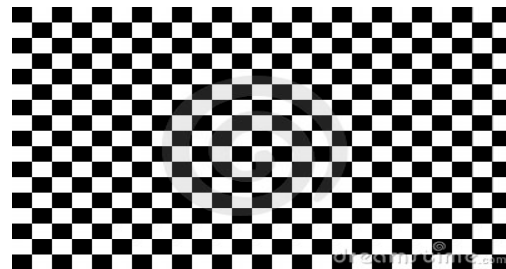
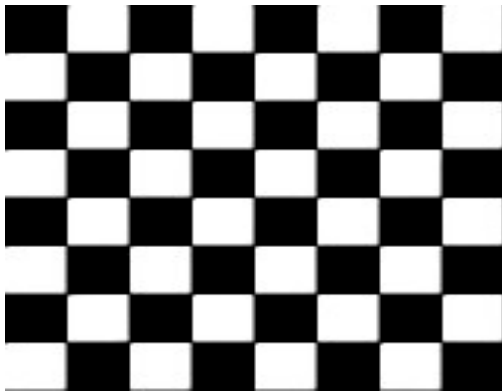
“A Markov chain wanders around the state space remembering only where it has been in the last period.”

Why Is this Useful?

- ◇ This “short-term” memory property is very useful because when the chain eventually finds the region of the state space with highest density, it will wander around there producing a sample that is only modestly nonindependent.
- ◇ If this is the posterior region, then we can use these “empirical” values as legitimate posterior sample values.
- ◇ Thus difficult posterior calculations can be done with MCMC by letting the chain wander around “sufficiently long”, thus producing summary statistics from recorded values.
- ◇ Sounds simple.

How Does it Move?

- ▶ How does the Markov chain decide to move?
- ▶ Define the *transition kernel*, K , as a general mechanism for describing the probability of moving to some other specified state based on the current chain status.
- ▶ $K(\theta, A)$ is a defined probability measure for all θ points in the state space to the set $A \in \Theta$.
- ▶ So $K(\theta, A)$ maps potential transition events to their probability of occurrence.



What is a Discrete Space Markov Chain Kernel?

- ◇ When the state space is discrete, K is a matrix mapping, $k \times k$ for k discrete elements in A , where each cell defines the probability of a state transition from the first term to all possible states:

$$P_A = \begin{bmatrix} p(\theta_1, \theta_1) & p(\theta_1, \theta_2) & \dots & p(\theta_1, \theta_{k-1}) & p(\theta_1, \theta_k) \\ p(\theta_2, \theta_1) & p(\theta_2, \theta_2) & \dots & p(\theta_2, \theta_{k-1}) & p(\theta_2, \theta_k) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p(\theta_k, \theta_1) & p(\theta_k, \theta_2) & \dots & p(\theta_k, \theta_{k-1}) & p(\theta_k, \theta_k) \end{bmatrix}$$

where the row indicates where the chain is at this period and the column indicates where the chain is going in the next period.

- ◇ Each matrix element is a well-behaved probability, $p(\theta_i, \theta_j) \geq 0$, $\forall i, j \in A$. When the state space is continuous, then K is a conditional PDF: $f(\theta|\theta_i)$.
- ◇ Rows of P_A sum to one and define a conditional PMF since they are all specified for the same starting value and cover each possible destination in the state space: for row i : $\sum_{j=1}^k p(\theta_i, \theta_j)$.

What is a Continuous Space Markov Chain Kernel?

- ◇ When the state space is continuous, then K is a conditional PDF: $f(\theta|\theta_i)$.
- ◇ K is a conditional PDF: $f(\theta|\theta_i)$, meaning a properly defined probability statement for all $\theta \in A$, given some given current state θ_i .
- ◇ Continuous state space Markov chains have more involved theory; so it's often convenient to think about discrete Markov chains at first.

A Two State Markov Chain

- ◇ A two-dimensional state space: a discrete vote choice between two political parties, a commercial purchase decision between two brands, etc.
- ◇ Voters/consumers/predators/viruses/etc. who normally select θ_1 have an 80% chance of continuing to do so, and voters/consumers who normally select θ_2 have only a 40% chance of continuing to do so.
- ◇ The transition matrix P :

$$\text{current period} \left\{ \begin{array}{l} \theta_1 \\ \theta_2 \end{array} \right. \left[\begin{array}{cc} \overbrace{0.8 \quad 0.2}^{\text{next period}} \\ \underbrace{0.6 \quad 0.4} \end{array} \right].$$

A Two State Markov Chain

◇ Assign a starting point:

$$S_0 = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix};$$

◇ To get to the first state, we simply multiply the initial state by the transition matrix:

$$S_1 = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.7 & 0.3 \end{bmatrix} = S_1.$$

A Two State Markov Chain

◇ This series continues multiplicatively as long as we like:

$$\text{Second state: } S_2 = \begin{bmatrix} 0.7 & 0.3 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.74 & 0.26 \end{bmatrix}$$

$$\text{Third state: } S_3 = \begin{bmatrix} 0.74 & 0.26 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.748 & 0.252 \end{bmatrix}$$

$$\text{Fourth state: } S_4 = \begin{bmatrix} 0.748 & 0.252 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.7496 & 0.2504 \end{bmatrix}.$$

A Two State Markov Chain

- ◇ The **Big Picture**: Imagine that this stationary distribution was the articulation of some PMF or PDF that we could not analytically describe but would like to. If we could run this Markov chain sufficiently long we would eventually get the stationary distribution *for any point in the state space*.

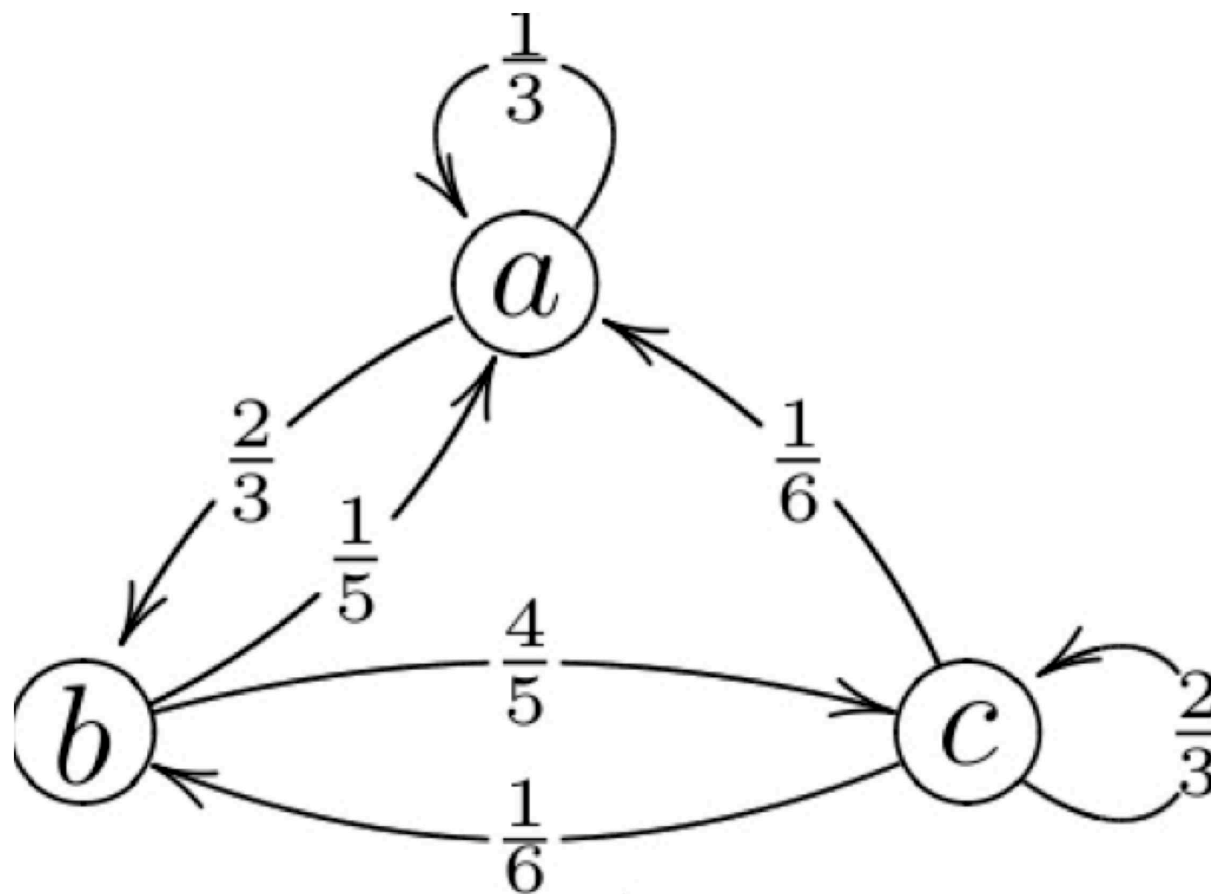
Actually, for this simple example we could solve directly for the steady state $S = [s_1, s_2]$ by stipulating:

$$\begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \end{bmatrix},$$

and solving the resulting two equations for the two unknowns.

- ◇ This operation of running a Markov chain until it reaches its stationary distribution is exactly the process employed in MCMC.

Diagrammatic Markov Chain



Another Simple Example

- ▶ Define the intended algorithm as some means of shuffling cards.
- ▶ The objective (stationary distribution) is a uniformly random distribution in the deck: for any given order to the stack, the probability of any one card occupying any one position is $1/52$.
- ▶ Algorithm: take the top card and insert it uniformly randomly at some other point in the deck, continue.
- ▶ Is this a Markov chain?
- ▶ What is the limiting distribution?
- ▶ See BTW, Bayer and Diaconis (1992).

Another Simple Example (cont.)

- ▶ For simplicity (without loss of generality), set $n = 3$ cards.
- ▶ Sample space ($3!$ elements): $\{[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]\}$.
- ▶ Transition kernel:

$$P = \begin{bmatrix} 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 1/3 & 0 & 1/3 \end{bmatrix}$$

- ▶ Starting point: $[1, 2, 3]$.
- ▶ Assume no periodicity, obviously it is irreducible and closed (therefore positive recurrent), therefore this transition kernel defines an ergodic Markov chain.

Another Simple Example (cont.)

R code:

```
P <- matrix(c(1/3,0,1/3,0,1/3,0,0,1/3,1/3,0,1/3,0,1/3,0,0,1/3,
             1/3,0,0,1/3,0,1/3,0,1/3,0,1/3,1/3,0,0,1/3,0,1/3,0,1/3),nrow=6)
```

P

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	0.3333333	0.0000000	0.3333333	0.3333333	0.0000000	0.0000000
[2,]	0.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[3,]	0.3333333	0.3333333	0.3333333	0.0000000	0.0000000	0.0000000
[4,]	0.0000000	0.0000000	0.0000000	0.3333333	0.3333333	0.3333333
[5,]	0.3333333	0.3333333	0.0000000	0.0000000	0.3333333	0.0000000
[6,]	0.0000000	0.0000000	0.3333333	0.3333333	0.0000000	0.3333333

Another Simple Example (cont.)

```
MC.multiply <- function(P.in,N) {  
  S <- c(1,0,0,0,0,0)%*%P.in  
  for (i in 2:N) {  
    S <- S%*%P.in  
    print(S)  
  }  
}
```

```
MC.multiply(P,15)
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  
[1,] 0.2222222 0.1111111 0.2222222 0.2222222 0.1111111 0.1111111  
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  
[1,] 0.1851852 0.1481481 0.1851852 0.1851852 0.1481481 0.1481481  
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  
[1,] 0.1728395 0.1604938 0.1728395 0.1728395 0.1604938 0.1604938
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1687243 0.1646091 0.1687243 0.1687243 0.1646091 0.1646091
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1673525 0.1659808 0.1673525 0.1673525 0.1659808 0.1659808
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1668953 0.1664380 0.1668953 0.1668953 0.1664380 0.1664380
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1667429 0.1665905 0.1667429 0.1667429 0.1665905 0.1665905
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1666921 0.1666413 0.1666921 0.1666921 0.1666413 0.1666413
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1666751 0.1666582 0.1666751 0.1666751 0.1666582 0.1666582
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1666695 0.1666638 0.1666695 0.1666695 0.1666638 0.1666638
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1666676 0.1666657 0.1666676 0.1666676 0.1666657 0.1666657
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.1666670 0.1666664 0.1666670 0.1666670 0.1666664 0.1666664
```


[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 0.1666668 0.1666666 0.1666668 0.1666668 0.1666666 0.1666666

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 0.1666667 0.1666666 0.1666667 0.1666667 0.1666666 0.1666666

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 0.1666667 0.1666666 0.1666667 0.1666667 0.1666666 0.1666666

Data Exercise 9: Coding a Markov Chain for Convergence Analysis

- ▶ You have seen the code for the playing cards example of Markov chain behavior.
- ▶ Now go back to the simple two-state example (toothpaste, voting, etc.) and write a function to run this chain.
- ▶ Analyze and describe the time to convergence. This should be short.
- ▶ Try different starting points. Does it matter?

Marginal Distributions

- ▶ We want the *marginal* distribution at some step m th from the transition kernel.
- ▶ For the discrete case the marginal distribution of the chain at the m step is obtained by inserting the current value of the chain, $\theta_i^{[m]}$, into the row of the transition kernel for the m^{th} step, p^m :

$$\pi^m(\theta) = [p^m(\theta_1), p^m(\theta_2), \dots, p^m(\theta_k)].$$

- ▶ So the marginal distribution at the **first step** of discrete Markov chain is given by:

$$\pi^1(\theta) = p^1 \pi^0(\theta),$$

where π^0 is the initial starting value assigned to the chain and $p^1 = p$ is a transition matrix.

Markov Chain Properties: Homogeneity

▶ *Homogeneity*

- ▶ A Markov chain is said to be *homogeneous* at step m if the transition probabilities at this step do not depend on the value of m .
- ▶ With non-homogeneous Markov chains the transition matrix is not constant but a function of time.
- ▶ These are useful for modeling probabilistic decay processes, but not for our purposes.
- ▶ The samplers that we use/pick are known to obtain the property, subject to machine implementation.
- ▶ Counter-example: at the starting point a chain cannot be homogeneous since the marginal distribution for the first step is clearly not independent of the initial values that are hand-picked.

Markov Chain Properties: States

► *Some Characteristics of States:*

▷ A state is *absorbing* if once the chain enters this state it cannot leave: $p(A, A^c) = 0$.

▷ A state is *transient* if the probability of the chain not returning to this state is non-zero:
 $1 - p(A, A) > 0$. Also, phrases as *a finite number of visits in infinite time*.

▷ State A is *closed* to state B if a Markov chain on A cannot reach B : $p(A, B) = 0$. State A is closed in the general sense if it is absorbing.

Markov Chain Properties: Irreducibility

► *Irreducibility*

- ▷ A state is *irreducible* if for every θ_i and θ_j in this state, the two sub-states “communicate.”
- ▷ A Markov chain is *irreducible* on A if every reached point or collection of points (necessarily a subspace in the continuous case) can be reached from every other reached point or collection of points:

$$p(\theta_i, \theta_j) \neq 0, \quad \forall \theta_i, \theta_j \in A.$$

- ▷ A convenient way to remember the principle behind irreducibility is the notion that you could reduce the set if you wanted to, but that *you do not want to* because then there will be points that cannot be reached from other points.

Markov Chain Properties: Recurrence

► *Recurrence*

- ▷ *If a state is closed, discrete or continuous but finite, and irreducible, then this state and all sub-spaces within this subspace are recurrent.*
- ▷ Markov chains operating on recurrent state spaces are recurrent.
- ▷ Recurrence is good: a chain that moves into a recurrent state, it stays there forever and visits every subspace infinitely often.
- ▷ Formal definition:
A irreducible Markov chain is called *recurrent* with regard to a given state, A , which is a single point or a defined collection of points (required for the bounded-continuous case), if the probability that the chain occupies A infinitely often over unbounded time is nonzero.
- ▷ For unbounded continuous state spaces we need a slightly more complicated form Harris recurrence.

Markov Chain Properties: Linkages

► *Recurrence and Irreducibility*

- ▷ The union of a set of recurrent states (nonempty, and bounded or countable) or Harris recurrent states is a new state that is closed and irreducible.
- ▷ The linkage between recurrence and irreducibility is important in defining a subspace that captures a Markov chain and at the same time assures that this Markov chain will explore all of the subspace.
- ▷ **Important result:**
 - Given a Markov chain on continuous state space,
 - when the chain wanders into a closed, irreducible set of Harris recurrent states,
 - it stays there forever and visits every single sub-state (region) with probability one.

Markov Chain Properties: Stationarity

► *Stationarity*

- Define $\pi(\theta)$ as the stationary distribution of the Markov chain for θ on the state space A .
- Recall that $p(\theta_i, \theta_j)$ is the probability that the chain will move from θ_i to θ_j at some arbitrary step t ,
- and $\pi^t(\theta)$ is the corresponding marginal distribution.
- The stationary distribution satisfies:

$$\sum_{\theta_i} \pi^t(\theta_i) p(\theta_i, \theta_j) = \pi^{t+1}(\theta_j) \quad \text{Discrete case}$$

$$\int \pi^t(\theta_i) p(\theta_i, \theta_j) d\theta_i = \pi^{t+1}(\theta_j) \quad \text{Continuous case.}$$

meaning that application of the kernel to the current distribution probabilities returns the same probabilities.

Markov Chain Properties: Periodicity

► *Periodicity*

- ▷ The *period* of a Markov chain is the length of time required to repeat an identical cycle of chain values.
- ▷ Actually, periodicity is bad and we want an *aperiodic* Markov chain: the repeating length is the trivial value of one.
- ▷ The problem with periodicity is that it destroys the probabilistic property of the transition kernel.
- ▷ Update of our status: we have recurrence to make sure that we revisit states infinitely often in the limit, we have stationarity to make sure that eventually we will obtain constancy of the probability structure, but we don't have any guarantee about what kind of distribution we obtain in this limit.
- ▷ Caveat: all Markov chains run on computers are actually periodic (“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.” –John von Neumann [1903-1957])

Markov Chain Properties: Ergodicity

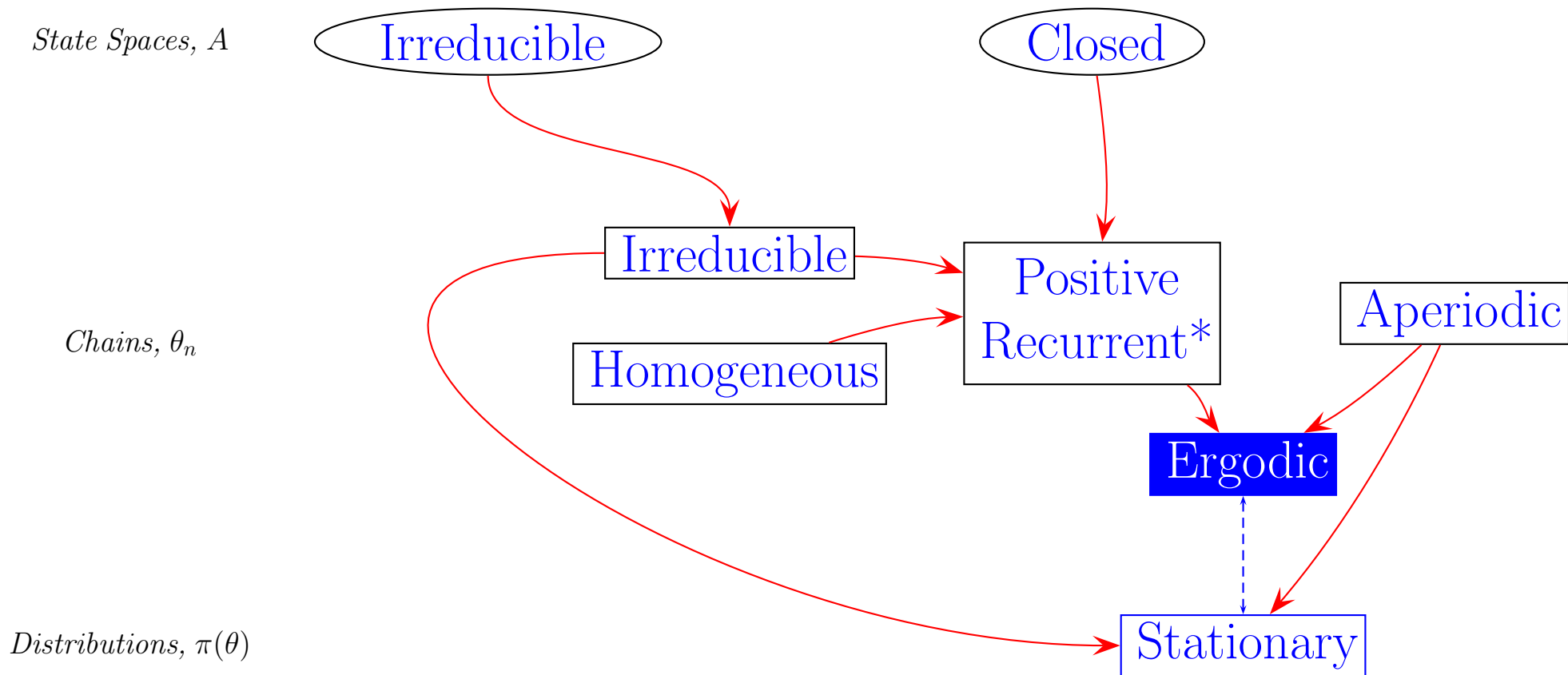
- ▶ If a chain is positive Harris recurrent (unbounded continuous case) or recurrent (discrete/bounded continuous case), and aperiodic, then we call it *ergodic*. Ergodic Markov chains have the property:

$$\lim_{n \rightarrow \infty} K^n(\theta_i, \theta_j) = \pi(\theta_j),$$

for all θ_i , and θ_j in the subspace

- ▶ What this means: *once a specified chain is determined to have reached its ergodic state, sample values behave as if they were produced by the posterior of interest from the model.*
- ▶ The *ergodic theorem* is the equivalent of the strong law of large numbers but for Markov chains, since it states that any specified function of the posterior distribution can be estimated with samples from a Markov chain in its ergodic state because averages of sample values give strongly consistent parameter estimates.

Markov Chain Properties



*Discrete or bounded continuous: $E[\eta_A] = \infty \forall \theta \in A$. Continuous: $P(\theta_n \in A) = 1, \forall A \in H$ where $P > 0$ (Harris). Positive recurrence: mean return time to A is bounded.

Markov Chain Properties

► *Consequences of Ergodicity*

- ▷ Suppose $\theta_{i+1}, \dots, \theta_{i+n}$ are n (not necessarily consecutive) values from a Markov chain that has reached its ergodic distribution, a statistic of interest, $h(\theta)$, can be calculated empirically:

$$\hat{h}(\theta_i) = \frac{1}{n} \sum_{j=i+1}^{i+n} h(\theta_j) = \int h(\theta) d\pi(\theta)$$

This is the ergodic theorem.

- ▷ For finite quantities this converges almost surely:

$$p[\hat{h}(\theta_i) \rightarrow h(\theta), \text{ as } n \rightarrow \infty] = 1$$

- ▷ Even though Markov chain values, by their very definition, have serial dependence, the mean of the chain values provides a strongly consistent estimate of the true parameter.

Markov Chain Properties

► *Markov Chain CLT*

- ▷ With ergodicity we get the important theoretical result from the CLT.
- ▷ For a given empirical estimator $\hat{h}(\theta_i)$ with bounded limiting variance, we get:

$$\sqrt{n} \frac{\hat{h}(\theta_i) - h(\theta)}{\sqrt{\text{Var}(\hat{h}(\theta_i))}} \xrightarrow[n \rightarrow \infty]{} \mathcal{N}(0, 1)$$

since $\hat{h}(\theta_i) = \frac{1}{n} \sum_{j=i+1}^{i+n} h(\theta_j)$ is really a mean.

The Gibbs Sampler

◇ Recall the conditional exponential PDFs:

$$f(x|y) \propto y \exp[-xy], \quad f(y|x) \propto x \exp[-xy], \quad 0 < x, y < B < \infty.$$

where we want to describe the marginal distributions of x and y .

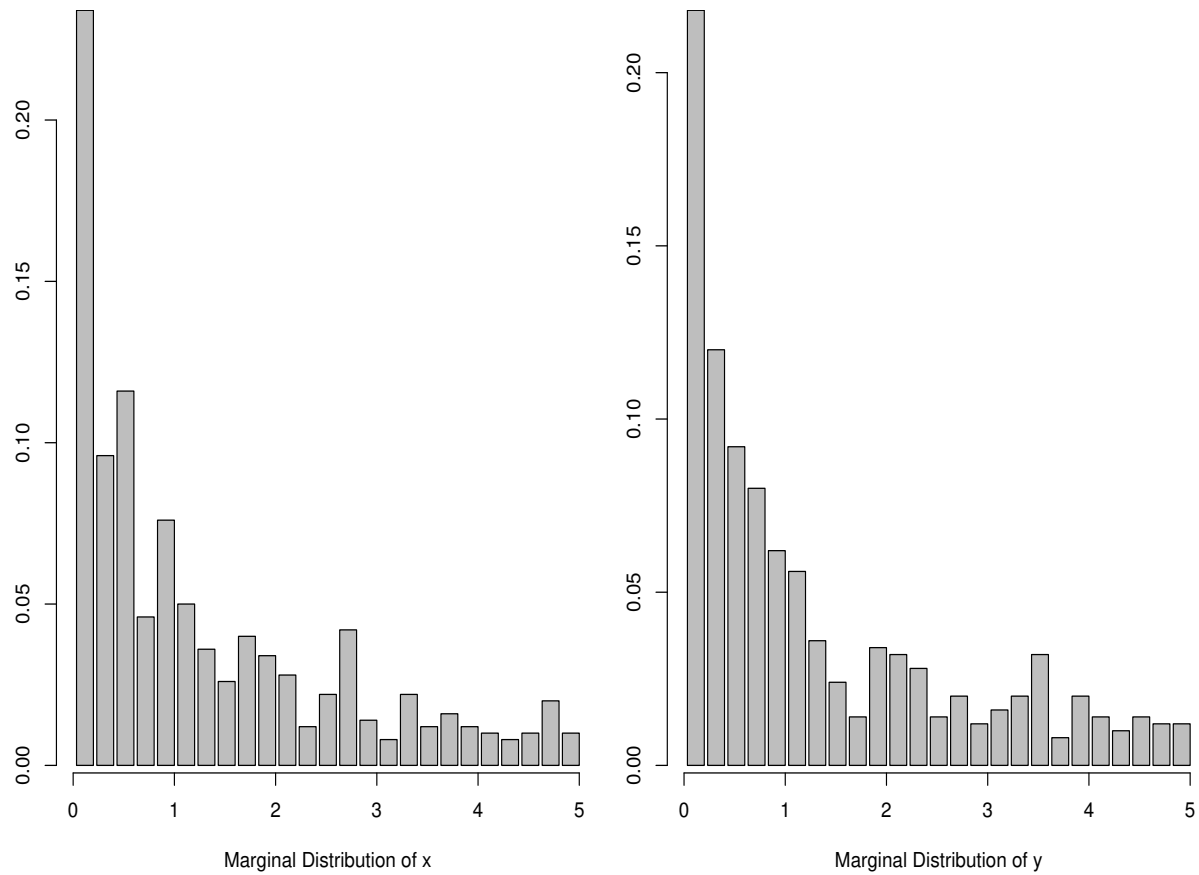
◇ For two parameters, x and y , this involves a starting point, $[x_0, y_0]$, and the cycles defined by drawing random values from the conditionals according to:

$$\begin{array}{ll} x_1 \sim f(x|y_0), & y_1 \sim f(y|x_1) \\ x_2 \sim f(x|y_1), & y_2 \sim f(y|x_2) \\ x_3 \sim f(x|y_2), & y_3 \sim f(y|x_3) \\ \vdots & \vdots \\ \vdots & \vdots \\ x_m \sim f(x|y_{m-1}), & y_m \sim f(y|x_m). \end{array}$$

The Gibbs Sampler, Conditional Exponential Distributions

```
B <- 5; k <- 15; m <- 500; x <- NULL; y <- NULL
while (length(x) < m) {
  x.val <- c(runif(1,0,B),rep((B+1),length=k))
  y.val <- c(runif(1,0,B),rep((B+1),length=k))
  for (j in 2:(k+1)) {
    while(x.val[j] > B) x.val[j] <- rexp(1,y.val[j-1])
    while(y.val[j] > B) y.val[j] <- rexp(1,x.val[j])
  }
  x <- c(x,x.val[(k+1)])
  y <- c(y,y.val[(k+1)])
}
```


The Gibbs Sampler (cont.)



Data Exercise 10: Coding a Gibbs Sampler

- ▶ Rerun the Gibbs sampler for the exponential distribution example for 100 iterations.
- ▶ Now run for 5,000 iterations and save the last 100 samples.
- ▶ Compare these two samples graphically and with summary statistics.
- ▶ How different are they?
- ▶ Why?

The Gibbs Sampler (cont.)

- ◇ The Gibbs sampler is a transition kernel created by a series of full conditional distributions.
- ◇ It is a Markovian updating scheme based on conditional probability statements.
- ◇ If the limiting distribution of interest is $\pi(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is an k length vector of coefficients to estimate, then the objective is to produce a Markov chain that cycles through these conditional statements moving toward and then around this distribution.
- ◇ The set of full conditional distributions for $\boldsymbol{\theta}$ are denoted Θ and defined by $\pi(\Theta) = \pi(\theta_i|\boldsymbol{\theta}_{-i})$ for $i = 1, \dots, k$, where the notation $\boldsymbol{\theta}_{-i}$ indicates a specific parametric form from Θ without the θ_i coefficient.

The Gibbs Sampler (cont.)

◇ Steps:

1. Choose starting values: $\boldsymbol{\theta}^{[0]} = [\theta_1^{[0]}, \theta_2^{[0]}, \dots, \theta_k^{[0]}]$.
2. At the j^{th} iteration, $j = 1, \dots, m$ complete the single cycle by drawing values from the k distributions given by:

$$\begin{aligned}
 \theta_1^{[j]} &\sim \pi(\theta_1 \mid \theta_2^{[j-1]}, \theta_3^{[j-1]}, \theta_4^{[j-1]}, \dots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
 \theta_2^{[j]} &\sim \pi(\theta_2 \mid \theta_1^{[j]}, \theta_3^{[j-1]}, \theta_4^{[j-1]}, \dots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
 \theta_3^{[j]} &\sim \pi(\theta_3 \mid \theta_1^{[j]}, \theta_2^{[j]}, \theta_4^{[j-1]}, \dots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
 &\vdots \\
 \theta_{k-1}^{[j]} &\sim \pi(\theta_{k-1} \mid \theta_1^{[j]}, \theta_2^{[j]}, \theta_3^{[j]}, \dots, \theta_{k-2}^{[j]}, \theta_k^{[j-1]}) \\
 \theta_k^{[j]} &\sim \pi(\theta_k \mid \theta_1^{[j]}, \theta_2^{[j]}, \theta_3^{[j]}, \dots, \theta_{k-2}^{[j]}, \theta_{k-1}^{[j]})
 \end{aligned}$$

3. Increment j and repeat until convergence.

Gibbs Sampler Theory

► Properties of the Gibbs sampler:

- Since the Gibbs sampler conditions only on values from the last iteration of its chain values, it clearly has the Markovian property.
- The Gibbs sampler has the true posterior distribution of the parameter vector as its limiting distribution: $\boldsymbol{\theta}^{[i]} \xrightarrow[i=1 \rightarrow \infty]{d} \boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$.
- The Gibbs sampler is a homogeneous Markov chain: the consecutive probabilities are independent of n , the current length of the chain.
- The Gibbs sampler converges at a geometric rate: the total variation distance between an arbitrary time and the point of convergence decreases at a geometric rate in time (t).
- The Gibbs sampler is ergodic.

The Gibbs Sampler, Realistic Example

- ▶ Consider a series of coal mine disasters over a 112-year history in the U.K. with relatively high disaster counts in the early era and relatively low disaster counts in the late era. Thus the question from a public policy perspective is when did improvements in technology and safety practices have an actual effect on the rate of serious accidents?

- ▶ The data, covering the years 1851 to 1962, are given (in sequential rows) by:

1851-1866	4 5 4 1 0 4 3 4 0 6 3 3 4 0 2 6	1-16
1867-1882	3 3 5 4 5 3 1 4 4 1 5 5 3 4 2 5	17-32
1883-1898	2 2 3 4 2 1 3 2 2 1 1 1 1 3 0 0	33-48
1899-1914	1 0 1 1 0 0 3 1 0 3 2 2 0 1 1 1	49-64
1915-1930	0 1 0 1 0 0 0 2 1 0 0 0 1 1 0 2	65-80
1931-1946	3 3 1 1 2 1 1 1 1 2 4 2 0 0 0 1	81-96
1947-1962	4 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1	97-112

The Gibbs Sampler, Realistic Example (cont.)

- ▶ This is an example of a *change-point problem* where the objective is to estimate when the parameterization of the underlying data-generation process changes.
- ▶ The statistical objective is to estimate the point where the Poisson parameter changes and obtain estimated values for this parameter before and after the change.
- ▶ Specifically, x_1, x_2, \dots, x_n are a series of count data where there exist the possibility of a change-point at some period, k , such that:

$$\begin{aligned}x_i|\lambda &\sim \mathcal{P}(\lambda) & i = 1, \dots, k \\x_i|\phi &\sim \mathcal{P}(\phi) & i = k + 1, \dots, n.\end{aligned}$$

So now there are three parameters to estimate: λ , ϕ , and k , where k plays a very different role than standard parameters.

The Gibbs Sampler, Realistic Example (cont.)

► Priors:

$$\lambda \sim \mathcal{G}(\alpha, \beta) \quad \phi \sim \mathcal{G}(\gamma, \delta) \quad k \sim \text{discrete uniform on } [1, 2, \dots, n],$$

where the prior parameters are assigned according to: $\alpha = 4$, $\beta = 1$, $\gamma = 1$, $\delta = 2$. Since the mean of a gamma distribution is the product of its parameters, this assignment of parameters roughly resembles the mean of the first 50% of the data ($\alpha\beta$), and the second 50% of the data ($\gamma\delta$).

► Joint posterior:

$$\begin{aligned} \pi(\lambda, \phi, k | \mathbf{y}) &\propto L(\lambda, \phi, k | \mathbf{y}) \pi(\lambda | \alpha, \beta) \pi(\phi | \gamma, \delta) \pi(k) \\ &= \left(\prod_{i=1}^k \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} \right) \left(\prod_{i=k+1}^n \frac{e^{-\phi} \phi^{y_i}}{y_i!} \right) \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \right) \\ &\quad \times \left(\frac{\delta^\gamma}{\Gamma(\gamma)} \phi^{\gamma-1} e^{-\delta\phi} \right) \frac{1}{n} \\ &\propto \lambda^{\alpha-1+\sum_{i=1}^k y_i} \phi^{\gamma-1+\sum_{i=k+1}^n y_i} \exp[-(k + \beta)\lambda - (n - k + \delta)\phi]. \end{aligned}$$

The Gibbs Sampler, Realistic Example (cont.)

► Conditional posterior distributions:

$$\lambda|\phi, k \sim \mathcal{G}\left(\alpha + \sum_{i=1}^k y_i, \beta + k\right) \quad \phi|\lambda, k \sim \mathcal{G}\left(\gamma + \sum_{i=k+1}^n y_i, \delta + n - k\right),$$

$$\begin{aligned} p(\mathbf{y}|k, \lambda, \phi) &= \left(\prod_{i=1}^k \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} \right) \left(\prod_{i=k+1}^n \frac{e^{-\phi} \phi^{y_i}}{y_i!} \right) \\ &= \left(\prod_{i=1}^k \frac{1}{y_i!} \right) \left(e^{-k\lambda} \lambda^{\sum_{i=1}^k y_i} \right) e^{-(n-k)\phi} \left(\prod_{i=k+1}^n \phi^{y_i} \right) \\ &= \left(\prod_{i=1}^k \frac{1}{y_i!} \right) \lambda^{\sum_{i=1}^k y_i} e^{-n\phi} e^{k(\phi-\lambda)} \left(\prod_{i=k+1}^n \phi^{y_i} \right) \left(\prod_{i=1}^k \frac{\phi^{y_i}}{\phi^{y_i}} \right) \\ &= \left(\prod_{i=1}^k \frac{e^{-\phi} \phi^{y_i}}{y_i!} \right) \left(e^{k(\phi-\lambda)} \left(\frac{\lambda}{\phi} \right)^{\sum_{i=1}^k y_i} \right) \\ &= f(\mathbf{y}, \phi) L(\mathbf{y}|k, \lambda, \phi). \end{aligned}$$

► This latter provides two functions, the first of which is free of k .

The Gibbs Sampler, Realistic Example (cont.)

- ▶ Since our objective is simply a full conditional statement for k , we will use only $L(\mathbf{y}|k, \lambda, \phi)$.
- ▶ Suppressing the conditioning on λ and ϕ for notational clarity only, apply Bayes' Law with a generic distribution for k , $p(k)$:

$$p(k|\mathbf{y}) = \frac{f(\mathbf{y}, \phi)L(\mathbf{y}|k)p(k)}{\sum_{\ell=1}^n f(\mathbf{y}, \phi)L(\mathbf{y}|k_{\ell})p(k_{\ell})} = \frac{L(\mathbf{y}|k)p(k)}{\sum_{\ell=1}^n L(\mathbf{y}|k_{\ell})p(k_{\ell})}.$$

- ▶ Here we took advantage of the discrete feature of k and summed over all possible values with the index ℓ .
- ▶ Due to the discrete prior, this simplifies even more such that with proportionality now $p(k|\mathbf{y}) \propto L(\mathbf{y}|k)$.
- ▶ So each iteration of the Gibbs sampler will calculate an n -length probability vector for k and draw a value accordingly.

R Code for the Realistic Example, (cont.)

```
bcp <- function(theta.matrix,y,a,b,g,d) {  
  n <- length(y)  
  k.prob <- rep(NA,length=n)  
  for (i in 2:nrow(theta.matrix)) {  
    lambda <- rgamma(1,a+sum(y[1:theta.matrix[(i-1),3]]),  
                    b+theta.matrix[(i-1),3])  
    phi <- rgamma(1,g+sum(y[theta.matrix[(i-1),3]:n]),  
                d+length(y)-theta.matrix[(i-1),3])  
    for (j in 1:n) k.prob[j] <- exp(j*(phi-lambda))*  
      (lambda/phi)^sum(y[1:j])  
    k.prob <- k.prob/sum(k.prob)  
    k <- sample(1:n,size=1,prob=k.prob)  
    theta.matrix[i,] <- c(lambda,phi,k)  
  }  
  return(theta.matrix)  
}
```

The Gibbs Sampler, Realistic Example (cont.)

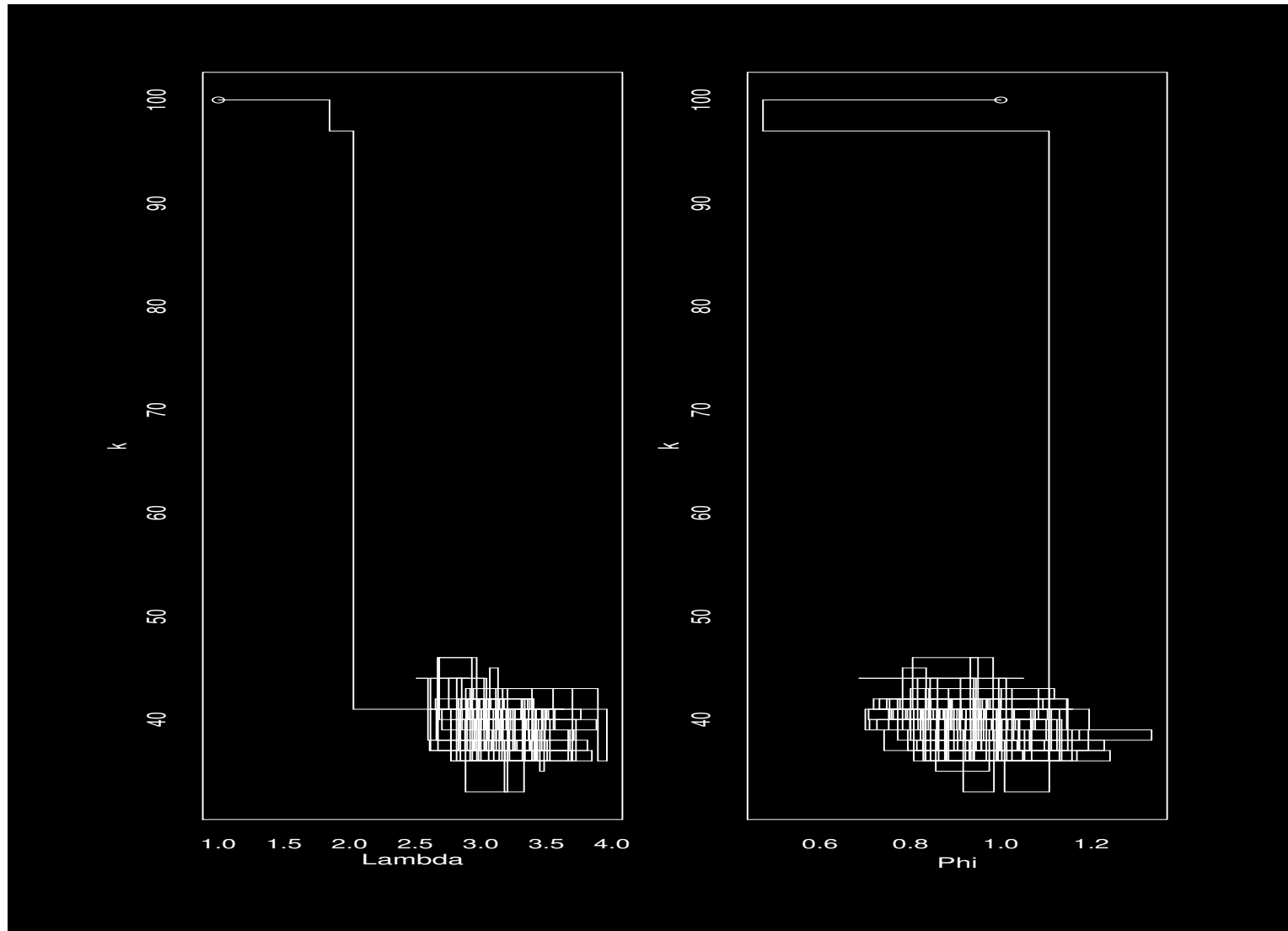
► Setup and Running:

```
coal.mining.disasters <- c(4,5,4,0,1,4,3,4,0,6,3,3,4,0,2,6,  
  3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,  
  2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,  
  1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,  
  0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,  
  3,3,1,1,2,1,1,1,1,2,4,2,0,0,1,4,  
  0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)  
num.reps <- 2000  
coal.mat <- matrix(NA,ncol=3,nrow=num.reps)  
coal.mat[1,] <- c(1,1,100)  
alpha <- 4; beta <- 1; gamma <- 1; delta <- 2  
coal.mat <- bcp(coal.mat,coal.mining.disasters,alpha,beta,gamma,delta)  
summary(coal.mat[1000:2000,])
```

The Gibbs Sampler, Realistic Example (cont.)

- ▶ This function is run for 2,000 iterations summarizing the last 1,000 chain values.

Quantile	λ	ϕ	k
Minimum	2.286	0.5975	32.00
First quartile	2.946	0.8616	39.00
Median	3.132	0.9359	40.00
Third quartile	3.339	1.0131	41.00
Maximum	4.337	1.4018	47.00
Mean	3.154	0.9447	39.77



Data Exercise 11: Modifying the Gibbs Sampler

- ▶ Rerun the Gibbs sampler for the coal mining example to replicate these results.
- ▶ Now modify the **R** code so that the sampler does not update each of the 3 parameters on every iteration. Instead it picks one of these only, uniformly randomly, and updates it alone.
- ▶ Do you get different results? Remember to run the chain long enough and delete the first half of the samples.

Returning to the Tobit Example for the Death Penalty, R Code For The Gibbs Sampler

```

B.mat <- matrix(summary(norr.tob)$coef, nrow=1) # STARTING POINTS
s.sq <- matrix(rep(10,6), nrow=1) # PRIOR on SIGMA^2 OF BETA
y.star <- Y # LATENT DATA START
for (i in mc.start:mc.stop) {
  for (j in 1:length(Y)) {
    if (Y[j] == 0) y.star[j] <- rtnorm(1, X[j,]%*%B.mat[i,], s.sq[i,], lower=-Inf, upper=0)
  }
  delta <- t(y.star - X%*%B.mat[i,])%*%(y.star - X%*%B.mat[i,])
  s.temp <- rep(Inf, ncol(s.sq)); while(!is.finite(sum(s.temp)))
    s.temp <- 1/rgamma(ncol(s.sq), (gamma0+length(Y))/2, (gamma1+delta)/2)
  s.sq <- rbind(s.sq, s.temp)
  B.hat <- solve( B0 + t(X)%*%X ) %*% (B0*beta0 + t(X)%*%y.star)
  B.var <- make.symmetric( solve( s.sq[(i+1)]^(-1)*B0 + s.sq[(i+1)]^(-1)*t(X)%*%X ) )
  B.mat <- rbind( B.mat, rmultinorm(1, B.hat, B.var, tol=1e-06) )
}
}

```


The Metropolis-Hastings Algorithm

- ▶ Another type of stochastic process that will help us estimate posterior quantities.

- ▶ Background:
 - ▷ The original work by Metropolis et al. postulated a two-dimensional enclosure with $n = 10$ molecular particles.

 - ▷ They sought to estimate the state-dependent total energy of the system at equilibrium.

 - ▷ Of course there is an incredibly large number of locations for the molecules in the system that must be accounted for and this number grows exponentially with time.

The Metropolis-Hastings Algorithm

► Continued:

- ▷ Their idea is to *simulate* this system probabilistically by generating moves that are more likely than others based on positions that are calculated using uniform probability generated candidate jump points.
- ▷ Therefore the simulation produces an estimated force based on a statistical, rather than deterministic, arrangement of particles.
- ▷ Obviously much easier.

Metropolis-Hastings Circa 1953

► Assumptions:

- ▷ We want to describe the posterior: $\pi(\theta)$, which is difficult.
- ▷ Candidate values will be generated from the distribution $q(\theta'|\theta)$ where $q(\cdot|\theta)$ a valid PDF for all values of θ , meaning:

$$\int q(\theta', \theta) d\theta = 1$$

- ▷ Also assume that this candidate generating (instrumental, jumping, or proposal) distribution is symmetric in its arguments:

$$q(\theta|\theta') = q(\theta'|\theta).$$

Otherwise the Markov chain is not irreducible.

- ▷ Note that the support of $\pi(\theta)$ and $q(\theta'|\theta)$ must be equivalent.

Metropolis-Hastings Circa 1953 (cont.)

► A single Metropolis iteration from the symmetric form has the following steps:

1. Sample θ' from $q(\theta'|\theta)$, where θ is the current location.

2. Sample u from $u[0 : 1]$.

3. If

$$a(\theta', \theta) = \frac{\pi(\theta')}{\pi(\theta)} > u$$

then accept θ' .

4. Otherwise keep θ .

► The result is:

$$\theta_0, \theta_1, \theta_2, \dots, \theta_n$$

where consecutive values are not necessarily unique.

Metropolis-Hastings Circa 1970

► Background:

- ▷ It turns out that the symmetric requirement of the instrumental distribution was an annoying restriction.
- ▷ Hastings (1970) as well as Peskun (1970) generalized the Metropolis et al. version by suggesting other jumping distributions such as the normal or Poisson.
- ▷ Question: can an asymmetric instrumental distribution work if there is some sort of compensation in the acceptance ratio?
- ▷ Yes (obviously), but there are conditions.

Metropolis-Hastings Circa 1970

► Generalizing:

- ▷ Define again $K(\theta', \theta)$ as the kernel of the Metropolis-Hastings algorithm going from θ to θ' .
- ▷ Sometimes $K(\theta', \theta)$ is labeled as $A(\theta', \theta)$ and called the *actual transaction function* from θ to θ' to distinguish it from $a(\theta', \theta)$.
- ▷ For Metropolis algorithms, there is a two-step process:

$$A(\theta', \theta) = p(\text{actually moving}) = p(\text{proposal}) \times p(\text{acceptance}) = q(\theta'|\theta) \times a(\theta', \theta)$$

- ▷ We now require that the transition kernel satisfy the *detailed balance equation*:

$$A(\theta', \theta)\pi(\theta') = A(\theta, \theta')\pi(\theta),$$

also called the *reversibility condition*.

Metropolis-Hastings Circa 1970

► Generalizing, cont.:

▷ Thus:

$$\frac{A(\theta', \theta)}{A(\theta, \theta')} = \frac{q(\theta|\theta') \pi(\theta')}{q(\theta'|\theta) \pi(\theta)}.$$

▷ This suggests, therefore, replacing:

$$a(\theta', \theta) = \frac{\pi(\theta')}{\pi(\theta)} > u \quad \text{with:} \quad a(\theta', \theta) = \frac{q(\theta|\theta') \pi(\theta')}{q(\theta'|\theta) \pi(\theta)} > u.$$

▷ Which means that we can use the same Metropolis-Hastings engine but replace symmetry with reversibility.

▷ Examples:

random walk chain, $\theta' = \theta + f(\tau)$

(where τ is some convenient random variable form).

independence chain, $\theta' = f(\tau)$

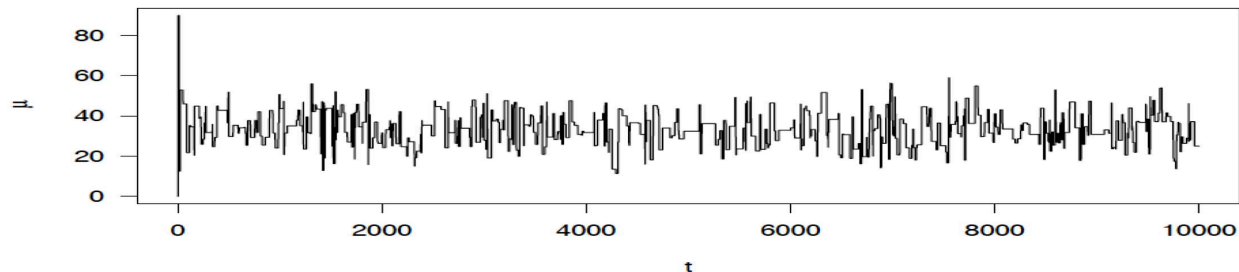
(ignores current position entirely in the candidate generation step).

The Metropolis-Hastings Transition Kernel

► Three possible events can occur:

1. sample a value of higher density and move with probability one,
2. sample a value of lower density but move anyway by drawing a small uniform random variable in Step 2 above,
3. or draw a uniform random variable larger than the ratio of posteriors and therefore stay in the same location.

► So unlike the Gibbs sampler, movement is not required on each iteration.



The Metropolis-Hastings Transition Kernel (cont.)

- ▶ **Key Point:** the choice for the jumping distribution variance really matters.
 - ▷ If the variance is too large, then the jumping distribution will be too wide relative to the target distribution and each successive step will move too far in some direction
 - ▷ If the jumping distribution variance is too small causing overly cautious small steps through the sample space and the chain converge slowly and mix poorly through the limiting distribution once we have converged.

Simple Metropolis-Hastings Example

- ▶ Suppose we want to simulate the bivariate normal distribution: $\mathcal{N} \left[\begin{array}{c|cc} 1 & & \\ \hline 2 & 1.0 & 0.9 \\ & 0.9 & 1.0 \end{array} \right]$.
- ▶ Lets disadvantage the algorithm somewhat by sampling from a bivariate t-distribution, starting with four overdispersed points in the sample space.
- ▶ A simple example, so it better work!

Simple Metropolis-Hastings Example

```
metropolis <- function(num.iter,I.mat) {  
  theta.matrix <- rbind(c(3,-2),matrix(NA,nrow=num.iter,ncol=2))  
  for (i in 2:nrow(theta.matrix)) {  
    theta.prime <- mvrnorm(1,theta.matrix[(i-1),],I.mat)/(sqrt(rchisq(2,5)/5))  
    a <- dmultinorm(theta.prime[1],theta.prime[2],c(0,0),I.mat)/  
      dmultinorm(theta.matrix[(i-1),1],theta.matrix[(i-1),2],c(0,0),I.mat)  
    if (a > runif(1)) theta.matrix[i,] <- theta.prime  
    else theta.matrix[i,] <- theta.matrix[(i-1),]  
  }  
  theta.matrix  
}
```

Simple Metropolis-Hastings Example

```
mcmc.mat <- metropolis(300,matrix(c(1,0.9,0.9,1),ncol=2))
par(mfrow=c(1,1),mar=c(3,2,2,2),oma=c(3,3,1,1),col.axis="white",col.lab="white",
    col.sub="white",col="white",bg="black")
plot(mcmc.mat,pch=1,cex=0.3,xlim=c(-3,3),ylim=c(-3,3))
lines(mcmc.mat,lwd=0.5,col="red")
axis(side=1,at=-3:3,NULL,col="white"); axis(side=2,at=-3:3,NULL,col="white")
```

Simple Metropolis-Hastings Example

