Advanced Bayesian Methods

Gabriel Katz

University of Exeter
Q-Step Centre
g.katz@exeter.ac.uk

Outline: Advanced Bayesian Computation

- The purpose of these notes is to take a closer look at the "nuts and bolts" of Bayesian inference
- We will build on "Introduction to Bayesian Data Analysis" by Andrei Zhirnov (Exeter, Q-Step)
- And focus on:
  1. The main algorithms used in modern Bayesian computation (Gibbs sampler and Metropolis-Hastings)
  2. Convergence criteria: how do we know when our Bayesian estimation "is ready"?
  3. Goodness-of-fit measures: how do we know whether our Bayesian model does a good job describing our data?
  4. Strategies to speed up execution time (integrating R with C++, Bayesian inference & High Performance Computing)

Recap: Bayes Theorem

- Let A and B be two events.
- Suppose we observe event B.
- What is the probability of observing A, given that we observed B?

Recap: Bayes Theorem

- Let A and B be two events.
- Suppose we observe event B.
- What is the probability of observing A, given that we observed B?
- Bayes Theorem:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{1}$$

- In (1):
  - $P(A|B)$ is the probability of A conditional on B
  - $P(B|A)$ is the conditional probability of B given A
  - $P(B)$ is the marginal probability of B

- In (1):
  - $P(A|B)$ is the probability of A conditional on B
  - $P(B|A)$ is the conditional probability of B given A
  - $P(B)$ is the marginal probability of B
- We can extend (apply) Bayes Theorem to random variables
  - this is the cornerstone of all Bayesian inference
  - because parameters **are random variables** within the Bayesian paradigm
  - which follow certain probability distributions

Bayes Theorem & Bayesian Inference

- Let
  - θ denote a parameter of interest (i.e., a regression coefficient, a variance parameter)
  - $f(\theta)$ be the probability distribution of θ
  - $f(\text{Data}|\theta)$ denote the sampling distribution of the data
    - i.e., the probability model followed by the data, given θ
    - $\approx$ likelihood function

- Applying Bayes Theorem:

$$f(\theta|\text{Data}) = \frac{f(\text{Data}|\theta)f(\theta)}{f(\text{Data})} \tag{2}$$

$$\left(\text{or: } f(\theta|\text{Data}) = \frac{f(\text{Data}|\theta)f(\theta)}{\int f(\text{Data}|\theta)f(\theta)} = \frac{f(\text{Data}|\theta)f(\theta)}{f(\text{Data})}\right)$$

- In Bayesian "parlance":
  1. $f(\theta)$ is the **prior distribution** of θ
     - i.e., before observing the data; what does the analyst "believe" about θ's distribution?
  2. $f(\text{Data}|\theta)$ is the probability (sampling) distribution of the data
  3. $f(\theta|\text{Data})$ is the **posterior distribution** of θ

- Since $f(\text{Data})$ does not depend on θ - i.e., it is a "constant" - we can write (2) as:

$$f(\theta|\text{Data}) \propto f(\text{Data}|\theta)f(\theta) \tag{3}$$

- (3) Gives us the "Bayesian mantra":

Posterior distribution $\propto$ **Likelihood** $\times$ **Prior distribution**

- Since $f(\text{Data})$ does not depend on θ - i.e., it is a "constant" - we can write (2) as:

$$f(\theta|\text{Data}) \propto f(\text{Data}|\theta)f(\theta) \tag{3}$$

- (3) Gives us the "Bayesian mantra":

  Posterior distribution $\propto$ **Likelihood** $\times$ **Prior distribution**

- Informally:
  - We start from a **prior distribution** for θ (before "observing" the data)
  - We "observe" the data
  - We update θ's distribution (i.e., update the **prior** once we observe the data) $\Rightarrow$ **posterior distribution**

- (3) is the fundamental relationship of Bayesian inference.

- The "whole purpose" of Bayesian inference: deriving the distribution of θ, given
  - the data (explanatory variables, X; dependent variable(s), Y)
  - the data model/likelihood, $f(\text{Data}|\theta)$
  - the **prior distribution** assumed for θ, before "observing" the data

Bayesian inference: the basic procedure

1. Specifying the data likelihood (the distribution of X and Y, given θ), $f(\text{Data}|\theta)$
2. Specifying the **prior distribution** for θ, $f(\theta)$
3. Deriving the **posterior distribution** of θ, $f(\theta|\text{Data})$
4. Once we have $f(\theta|\text{Data})$, we can summarize this distribution
   - e.g., compute the (posterior) mean, variance, median, quantiles, etc.

An example of Bayesian inference

- Suppose 2 candidates, A y B, are competing in an election.
- An opinion poll based on a representative sample was conducted days before the election:
  - 1,067 potential voters
  - 556 of which stated their intention to vote for A
  - 511 declared they would vote for B
- Based on these data, A hires a Bayesian researcher in order to assess his chances of winning the election
- Let's go over the 4 steps of Bayesian analysis in this case

Step 1: Specifying the data model

- Each survey participant has two choices (ignoring abstention):
  - vote for A
  - vote for B
- We can think of this as $n = 1067$ Bernoulli trials, where success="voting for A".
- Let p denote the probability of success, and Y_i the choice of individual $i = 1, 2, \ldots, 1067$, with:
  - $Y_i = 1$ if i would vote for A
  - $Y_i = 0$ if i would vote for B

- The sampling distribution of the data can be then written as:

$$f(\text{Data}|p) = \prod_{i=1}^{1067} p^{Y_i} (1-p)^{1-Y_i} \tag{4}$$

- The sampling distribution of the data can be then written as:

$$f(\text{Data}|p) = \prod_{i=1}^{1067} p^{Y_i} (1-p)^{1-Y_i} \tag{4}$$

- Based on the responses to the public opinion poll:

$$f(\text{Data}|p) = p^{556} (1-p)^{511} \tag{5}$$

- The **posterior distribution** of p will then be given by:

$$f(p|\text{Data}) \propto p^{556} (1-p)^{511} f(p)$$

Step 2: choosing the **prior distribution** for p

- A natural prior distribution for parameters representing rates or proportion is the **Beta distribution**
- In fact, the Beta distribution is a **conjugate prior** for parameters representing rates or proportions
- We say that $f(\theta)$ is a **conjugate prior** distribution when
  - $f(p|\text{Data})$ follows the same distribution as $f(\theta)$
- In our application, if $f(p) \sim$ Beta

$$f(p|\text{Data}) = p^{556}\,(1-p)^{511} \times f(p) \sim \textit{Beta}$$

The Beta distribution

- Let p be a random variable, $0 < p < 1$
- If p follows a Beta distribution, its density function is:

$$f(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{6}$$

with parameters α y β, and $\Gamma()$ a gamma function:

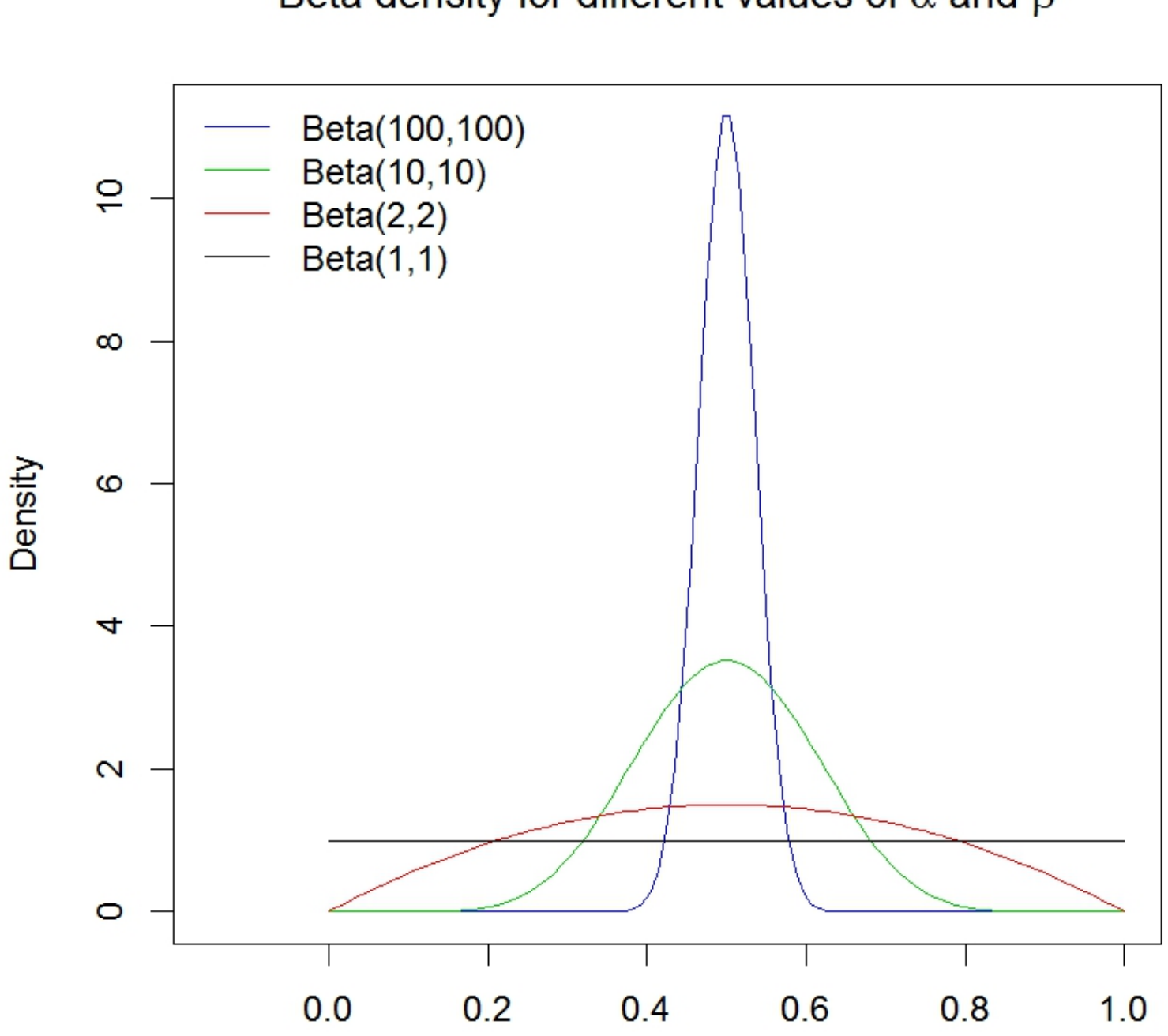
$$\Gamma(n) = \int_0^\infty u^{n-1} exp(-u) du, n > 0$$

with

- $\Gamma(n) = (n-1)!$ for n a positive integer

- The characteristics and central moments of a Beta random variable depend on α y β

- The characteristics and central moments of a Beta random variable depend on α y β
  - $E(p) = \frac{\alpha}{(\alpha+\beta)}$
  - $Var(p) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
- R: "Beta Distribution.R"


Beta density for different values of α and β
Beta(100,100)
Beta(10,10)
Beta(2,2)
Beta(1,1)
Density
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0

- The plot of the Beta density shows that the Bayesian analyst must choose
  - not only the prior distribution for the parameters (e.g., p)
  - but also the parameters of this *prior distribution* (in this case, α & β)
- The parameters of a *prior distribution* are known as the **hyperparameters**
- These **hyperparameters** will influence the "weight" that the *prior distribution* has on the *posterior distribution* of p

- For instance, suppose the analyst has no prior information about the possible result of the election.

- From (6), it follows that if the analyst chooses the following values for the **hyperparameters**:

  - $\alpha = 1$
  - $\beta = 1$

  the **prior distribution** for p will be:

$$f(p) = \frac{\Gamma(1+1)}{\Gamma(1)\Gamma(1)} p^{1-1}(1-p)^{1-1} = \frac{2!}{0!0!} p^0(1-p)^0$$

$$\Rightarrow f(p) \propto p^0(1-p)^0 = 1$$

- And therefore, the **posterior distribution** for p will be:

$$f(p|\text{Data}) \propto p^{556}(1-p)^{511} \times 1$$

$$\Rightarrow f(p|\text{Data}) \propto p^{556}(1-p)^{511}$$

- In other words, the **prior distribution** adds no information to the **posterior**
  - The **posterior distribution** will be completely determined by the data
- This type of **prior distribution** is known as **vague** or **weakly informative** prior
  - when **vague** priors are used, Bayesian inference is $\approx$ maximum likelihood inference

- At the other extreme, suppose the analyst also has information from previous opinion polls:

Poll	n	Votes for A	Votes for B
Last month	685	346	339
2 months ago	637	312	325
3 months ago	628	284	344
Total	1,950	942	1,008

- The analyst could choose to incorporate this information in the **prior** $f(p)$.

- For instance:

$$f(p) = \frac{\Gamma(1950)}{\Gamma(942)\Gamma(1008)} p^{942-1}(1-p)^{1008-1}$$

- And thus the **posterior distribution** would become

$$f(p|\text{Data}) \propto p^{556}(1-p)^{511} \times p^{941}(1-p)^{1007}$$

$$\Rightarrow f(p|\text{Data}) \propto p^{1497}(1-p)^{1518} \tag{7}$$

Step 3: Deriving the posterior distribution for p

- In sum, we arrived at two possible **posterior distributions** for p
- With a **vague prior Beta distribution**:

$$f(p|\text{Data}) = p^{556}(1-p)^{511}$$

- With an **informative Beta prior distribution**:

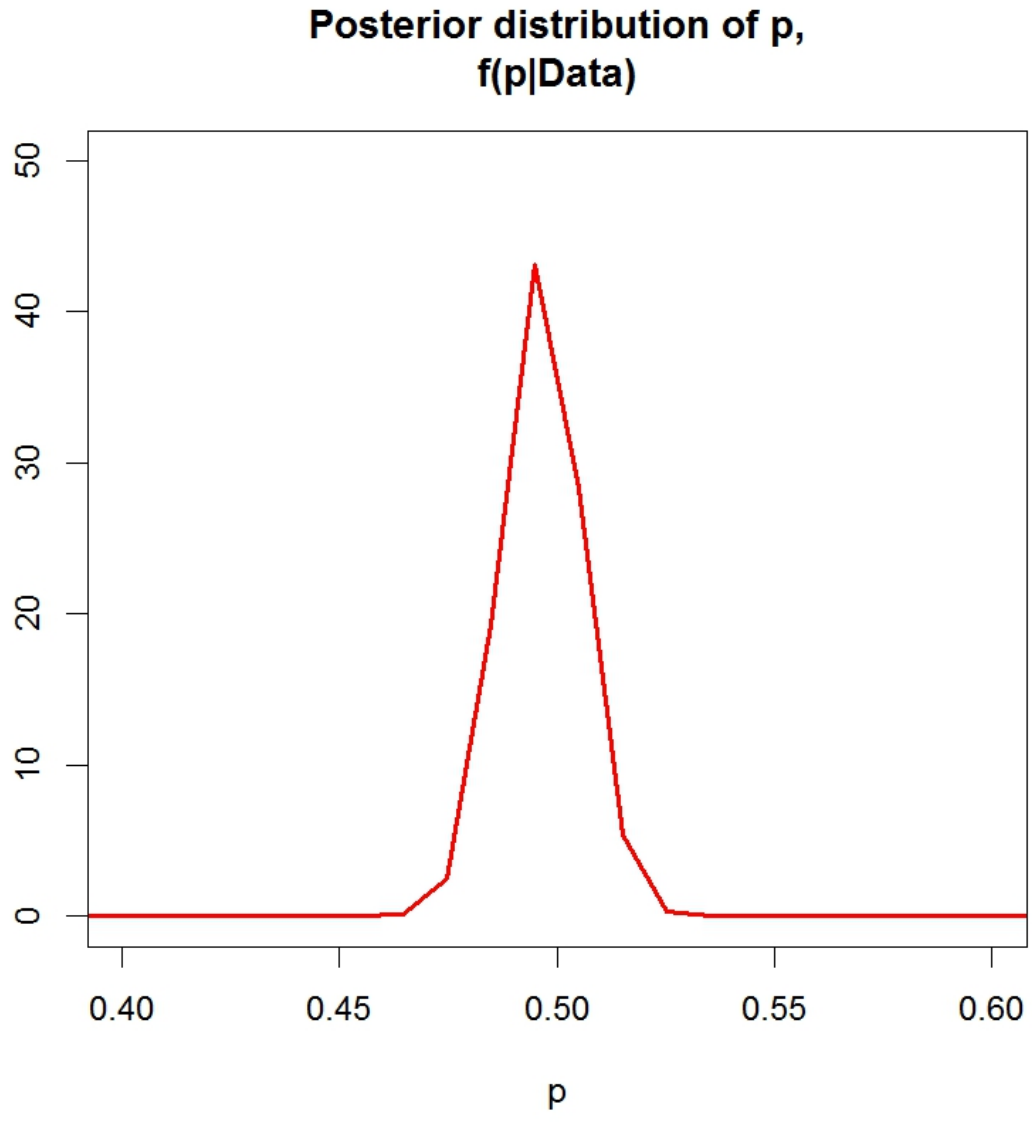
$$f(p|\text{Data}) = p^{1497}(1-p)^{1518}$$

- In both cases, we have a **Beta posterior**
  - because we chose a **conjugate prior distribution**

- The choice of **hyperparameters** α and β will determine whether $f(p|\text{Data})$ will be affected
    - primarily by the data under consideration (if the prior is **vague** or **weakly informative**)
    - by the data and prior information (if the **prior distribution** is **informative**)

Step 4: Summarizing $f(p|\text{Data})$

- Assuming that $f(p) \propto Beta(984, 1008)$
  - & given the data model $f(\text{Data}|p)$
  - we have: $f(p|\text{Data}) \propto Beta(1498, 1519)$
- So ... what is the "estimate" of p?
  - Bayesian inference does not lead to "estimates" ... it leads to (posterior) probability distributions for parameters like p
- R: "p - Posterior Distribution.R"


Posterior distribution of p,
f(p|Data)
0
10
20
30
40
50
0.40
0.45
0.50
0.55
0.60
p

- So, how do we summarize $f(p|\text{Data})$?

- So, how do we summarize $f(p|\text{Data})$?
- We can compute the expected value of p:

$$E(p) = \frac{\alpha}{\alpha + \beta} = \frac{1498}{14998 + 1519} = 0.497 \tag{8}$$

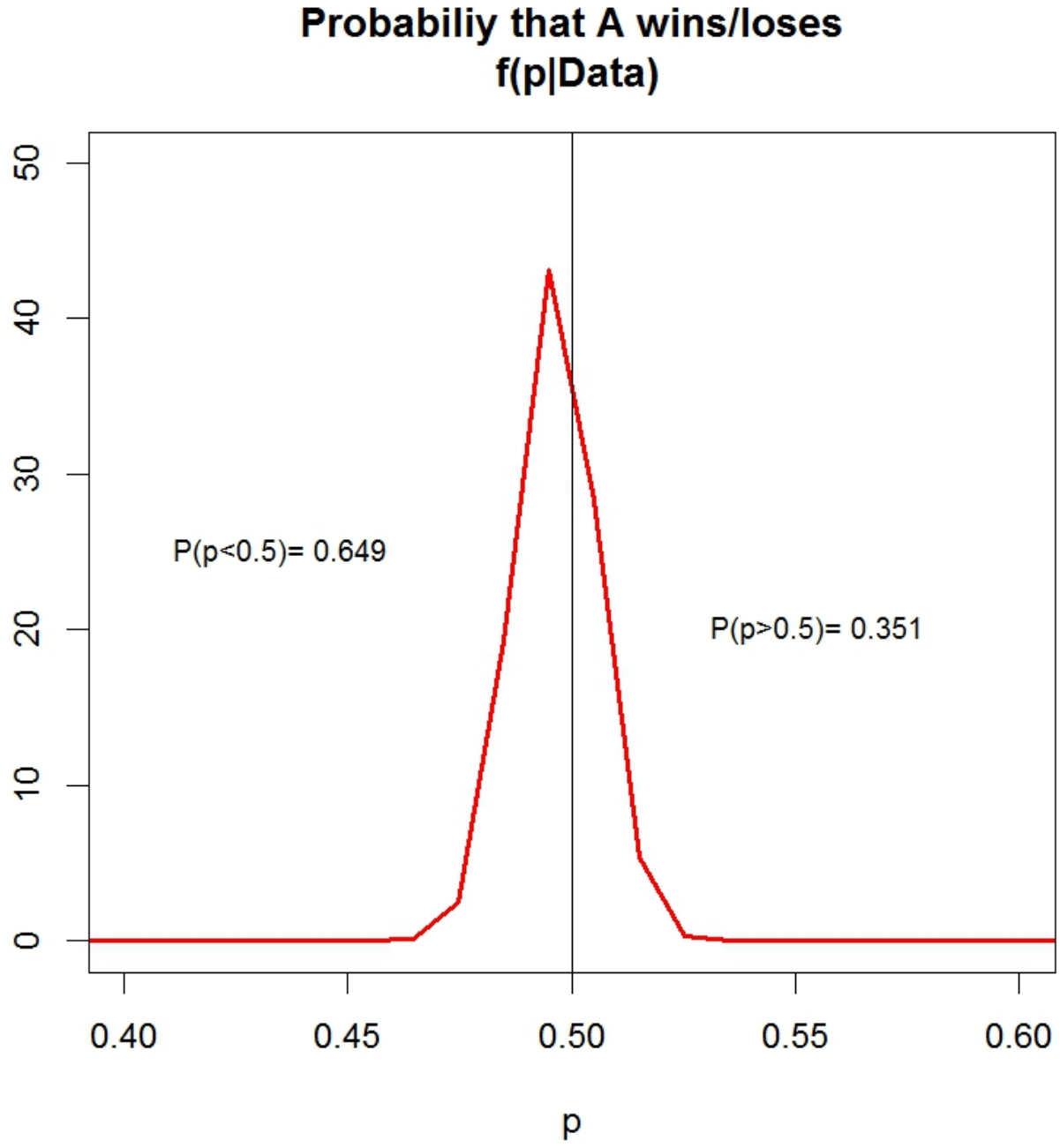
- That is, the expected proportion of votes for A is 49.7%
- We can also compute a 95% **credibility interval** for p
  - R: "Posterior summaries for p.R"

- We can also compute the probability that A wins/loses the election

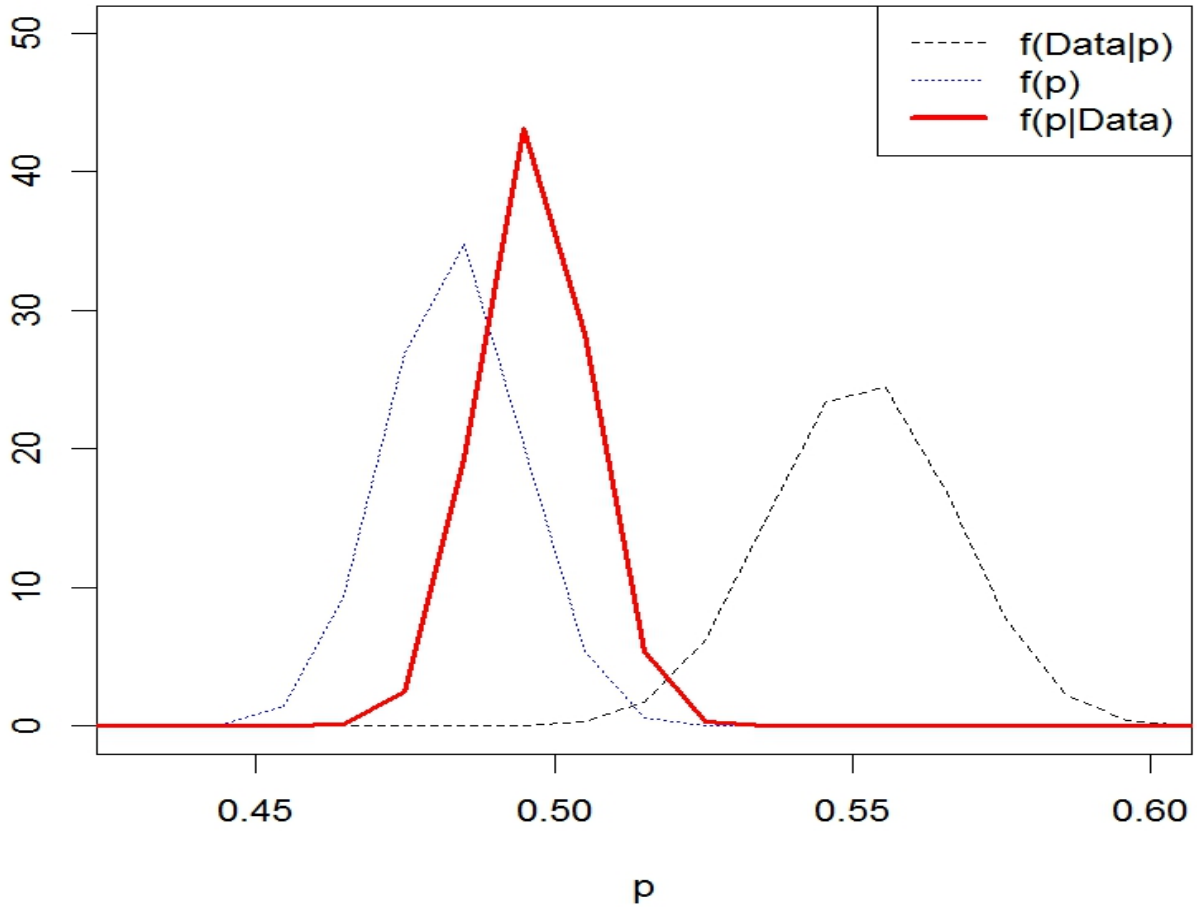
$$P(A \text{ wins}) = P(p \geq 0.5) = 0.351$$

$$P(A \text{ loses}) = P(p \leq 0.5) = 0.649$$

- R: "Posterior summaries for p.R"

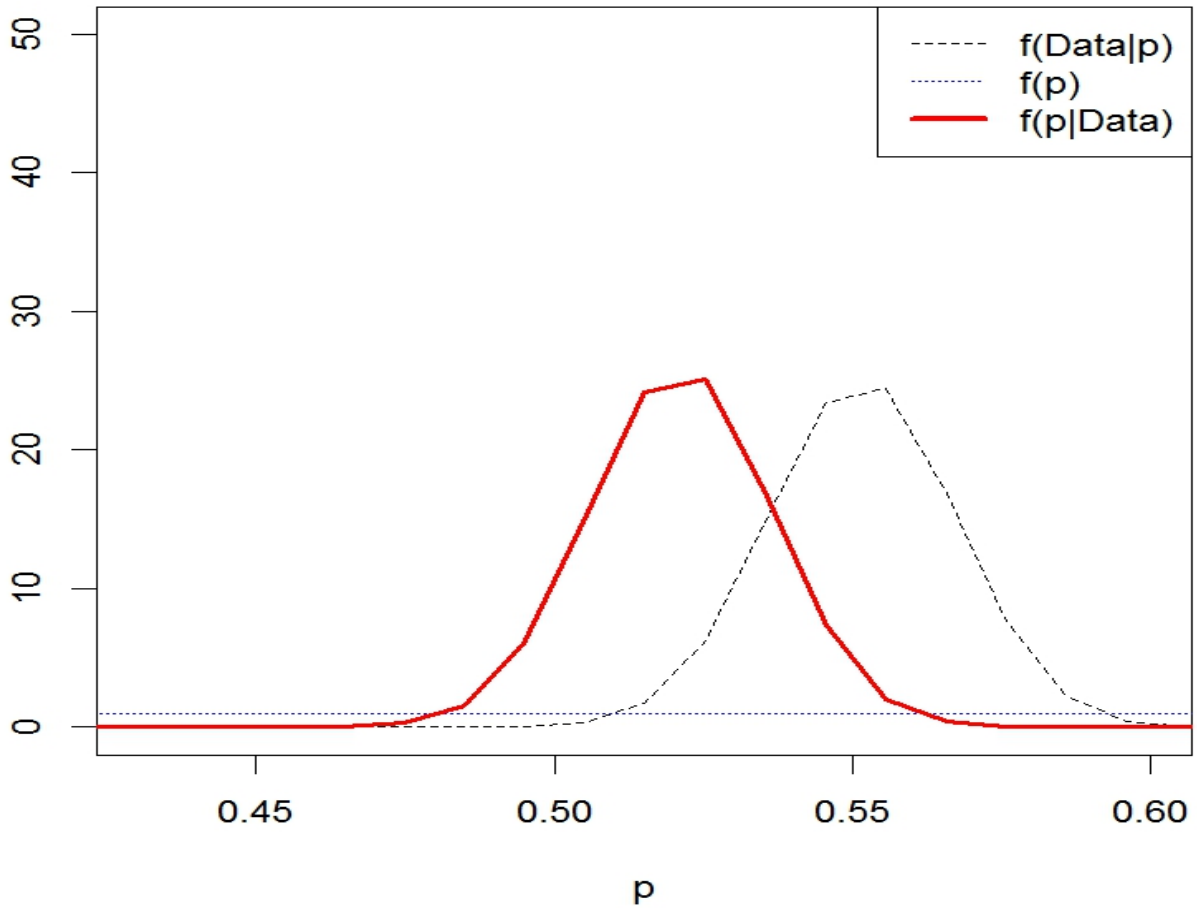

Probabiliy that A wins/loses
f(p|Data)
P(p<0.5)= 0.649
P(p>0.5)= 0.351
0
10
20
30
40
50
0.40
0.45
0.50
0.55
0.60
p

Side note: Comparison between $f(p)$, $f(\text{Data}|p)$, y $f(p|\text{Data})$



- Note that $f(p|\text{Data})$ is a "compromise" between
  - $f(p)$
  - and $f(\text{Data}|p)$
- When we use an **informative** $f(p)$, $f(p|\text{Data})$ is "closer" to $f(p)$ than to $f(\text{Data}|p)$
  - because $f(p)$ carries more weight than $f(\text{Data}|p)$
  - 1,950 respondents in previous polls
  - but only 1,067 in the poll under study
- The situation would have been different with a **"vague"** $f(p)$

Comparison: $f(p)$, $f(\text{Data}|p)$, y $f(p|\text{Data})$, **vague** $f(p)$



- In this case:
  - $E(p) = 0.55$
  - 95% creditibility interval: $[0.49, 0.55]$
  - $Prob(A \quad \text{wins}) := 0.915$
- $\Rightarrow$ Critical role of **prior distributions** in Bayesian inference

A first approximation to simulation-based inference

- In the previous exercise, we derived $f(p|\text{Data}) \sim Beta(1498, 1519)$

- Based on this, we could compute $E(p|\text{Data})$ applying the formula for the expectation of a Beta random variable

$$E(p|\text{Data}) = \frac{\alpha}{\alpha + \beta} \tag{9}$$

A first approximation to simulation-based inference

- In the previous exercise, we derived $f(p|\text{Data}) \sim Beta(1498, 1519)$

- Based on this, we could compute $E(p|\text{Data})$ applying the formula for the expectation of a Beta random variable

$$E(p|\text{Data}) = \frac{\alpha}{\alpha + \beta} \tag{9}$$

- (9) follows from

$$E(p|\text{Data}) = \int_0^1 p \times \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} dp$$

- Similarly, we could compute $P(p > 0.5|\text{Data})$ from

$$1 - P(p < 0.5|\text{Data}) = 1 - \int_0^{0.5} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1} dt$$

- Similarly, we could compute $P(p > 0.5|\text{Data})$ from

$$1 - P(p < 0.5|\text{Data}) = 1 - \int_0^{0.5} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1} dt$$

- Now, suppose we could not "solve" these integrals
  - How could we compute $E(p|\text{Data})$, $P(p > 0.5|\text{Data})$?
  - or other quantities of interest like the median, the credible intervals, etc.?

- Similarly, we could compute $P(p > 0.5|\text{Data})$ from

$$1 - P(p < 0.5|\text{Data}) = 1 - \int_0^{0.5} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1} dt$$

- Now, suppose we could not "solve" these integrals
  - How could we compute $E(p|\text{Data})$, $P(p > 0.5|\text{Data})$?
  - or other quantities of interest like the median, the credible intervals, etc.?

- "Bayesian answer": Simulations!

- We can generate S values from a $Beta(1498, 1519)$ distribution
- Let's write these S values as $p^1, p^2, \ldots, p^S$
- We can compute $E(p|\text{Data})$ as

$$\int p\, f(p|\text{Data}) \approx \frac{\sum_{s=1}^{S} p^s}{S} \tag{10}$$

- Similarly, we can compute $Var(p)$ as

$$\int (p - E(p|\text{Data}))^2 f(p|\text{Data}) \approx \frac{\sum_{s=1}^{S} (p^s - \bar{p})^2}{S} \tag{11}$$

with $\bar{p}$ obtained from (10)

- Likewise, we can compute any other quantity of interest using simulations
    - $P(p > 0.5|\text{Data})$
    - credibility intervals, etc.

- R: "Simulations from a Beta posterior.R"

Monte Carlo Integration

- More generally, suppose we want to compute

$$E\Big(h(\theta)\Big) = \int h(\theta) f(\theta|\text{Data}) d\theta \tag{12}$$

  where:

  - θ is the parameter of interest
  - $h(\theta)$ is a function of θ

- Solving (12) analytically can be quite difficult

- But suppose we can generate S **independent values** from $f(\theta|\text{Data})$:

$$\theta^1, \theta^2, \ldots, \theta^s \sim f(\theta|\text{Data}) \tag{13}$$

- We can then compute $E\Big(h(\theta)\Big)$ as:

$$E\Big(h(\theta)\text{Data}\Big) \approx \bar{h} = \frac{\sum_{s=1}^{S} h(\theta^s)}{S} \tag{14}$$

- This is the principle behind Monte Carlo integration (simulation)
  - series of techniques to compute probabilities and moments based on values simulated from a probability distribution
  - instead of using "calculus"

Why does Monte Carlo integration (simulation) work?

- Under very general conditions, a (certain version) of the law of large numbers guarantees that

$$\frac{\sum_{s=1}^{S} h(\theta^s)}{S} \rightarrow E\Big(h(\theta)\Big) \tag{15}$$

  as $S \rightarrow \infty$

- Important: it is not n that "goes to" ∞
  - but rather S, the number of values simulated (generated) from $f(\theta|\text{Data})$

- With modern computers, it is very easy and cheap to generate large number of values S from $f(\theta|\text{Data})$

- Monte Carlo integration is thus a very general and powerful approach
- Any integral or sum can be expressed as the expectation of a random variable with resepect to a certain probability distribution
- In other words, we can compute expectations, variances, quantiles, etc. using Monte Carlo integration
- **As long as we can directly generate S independent values from $f(\theta|\text{Data})$**
- This is not always possible, though.

Markov chain Monte Carlo (MCMC) simulation

- In the previous example (two-candidate election), it was very easy to generate values from $f(\theta|\text{Data})$
  - or, more specifically, $f(p|\text{Data})$

  using a simple R command ("rbeta")

- In other cases, this is more difficult
  - e.g., models with multiple parameters
  - models for which $f(\theta|\text{Data})$ does not have a closed form

- So we need specific routines/algorithms to sample from $f(p|\text{Data})$

- For example, let's assume that

$$f(\theta|\text{Data}) \propto \frac{2\theta + 3}{40} \qquad 0 < \theta < 5 \qquad (16)$$

- There is no "command" or "canned routine" allowing us to generate values from (22).
  - "rbeta" in R is no longer useful for us
  - or any other random generation function available in R (or MATLAB, or Stata, or Python) for that matter
- In such circumstances, me need more "specialized" algorithms to generate samples from $f(\theta|\text{Data})$

- There are multiple Monte Carlo simulation methods (routines, algorithms) to deal with this sort of cases
  - e.g. importance sampling
  - rejection sampling
- Bayesian statistics relies on Markov chain Monte Carlo simulations (MCMC)

Basic idea behind MCMC simulations

- The denomination "Markov chain Monte Carlo (MCMC) refers to the two key "components" of the simulation techniques that conform the basis of modern Bayesian computation

  1. We want to "learn" about the **posterior distribution** of a parameter θ, $f(\theta|\text{Data})$, using Monte Carlo simulations
     - rather than analytically solving for the moments (expectation, variance, etc.) of $f(\theta|\text{Data})$
     - we are going to simulate values from $f(\theta|\text{Data})$
  2. Those values are going to be simulated form a **Markov chain**
     - because (15) holds in this case

Markov chain

- A sequence $X^0, X^1, X^2, \ldots$ of random variables
  - uni o multi-variate

  is a Markov chain if:

$$P(X^{t+1}|X^t, X^{t-1}, \ldots, X^0) = P(X^{t+1}|X^t) \ \forall s \in N \quad (17)$$

- That is, only the value(s) of X en t is (are) relevant for the distribution of X in $t+1$
- Under general conditions, the Markov chain converges - in distribution - to its **stationary distribution** as $t \to \infty$
  - regardless of the chain's starting point

- The conditions under which a Markov chain converges to its stationary distribution are rather "technical"

    1. **Irreductibility:** Any given state of X can be reached from any other state in a finite number of moves

    2. **Aperiodicity**: the chain does not cyclically return to a previous state

- Technical references and conditions under which they hold:
    - Geyer (1992), Besag y Green (1993), y referencias

- More important for our purposes: the conditions under which a Markov chain converges to its stationary distribution are quite general

    - for practical purposes, they hold for the vast majority of social science models

MCMC simulations to approximate $f(\theta|\text{Data})$

- Let's suppose that:
  - We replace X in the previous definition with a parameter θ
  - We replace the index t with s (simulations)
- The basic idea behind MCMC simulations is to "construct" a Markov chain such that:
  - starting from an initial, arbitrary value of θ, θ^0
  - we can generate a sequence of samples θ^s, each of which depends only on θ^{s-1}, $s = 1, \ldots, S$
  - and that sequence convergences to the stationary distribution $f(\theta|\text{Data})$
- The same idea generalizes to vectors of parameters Θ

Two main MCMC simulation techniques

- Most econometric/statistical models use in the social science involve several parameters, relatively "complex" posterior distributions, etc
  - impossible or tedious to solve analytically
- Modern Bayesian inferences resorts to two basic techniques (algorithms) to "build" Markov chains that converge to $f(\Theta|\text{Data})$
  1. Gibbs sampling:
  2. Metropolis-Hastings algorithm

1. Gibbs sampling:
   - the most basic algorithm in Bayesian inference
   - applied when it is impossible to simulate from the **(joint) posterior distribution** of the parameters
   - but we can "split" this joint posterior distribution into a series of simpler **conditional** distributions
   - from which it is easier to generate samples

2. Metrópolis-Hastings algorithm
   - generalization of Gibbs sampling
   - useful when it is impossible to generate values even from these **conditional posterior** distributions

Gibbs sampling

- Suppose we have a vector of parameters Θ, with K elements $\Theta = (\theta_1, \ldots, \theta_K)$
- Suppose it is impossible to draw samples from

$$f(\Theta|\text{Data}) = f(\theta_1, \theta_2, \ldots, \theta_K|\text{Data})$$

- However, suppose I can "split" the **joint posterior distribution** into a series of conditional distributions:

$$\begin{aligned} f(\Theta|\text{Data}) = & f(\theta_1|\theta_2, \ldots, \theta_K, \text{Data}) \\ & \times f(\theta_2|\theta_1, \theta_3, \ldots, \theta_K, \text{Data}) \cdots \times \\ & f(\theta_K|\theta_1, \ldots, \theta_{K-1}, \text{Data}) \end{aligned}$$

- Or, equivalently,

$$f(\Theta|\text{Data}) = f(\theta_1|\theta_{-1}, \text{Data}) \times f(\theta_2|\theta_{-2}, \text{Data}) \cdots \times f(\theta_K|\theta_{-K}, \text{Data})$$

  where θ_{-k}, $k = 1, \ldots, K$, denotes all the elements of Θ except for θ_k

- If:
  - We cannot draw samples from $f(\Theta|\text{Data})$
  - But can draw samples from $f(\theta_k|\theta_{-k}, \text{Data})$, $k = 1, \ldots, K$
  - $\Rightarrow$ I can use Gibb sampling to approximate $f(\Theta|\text{Data})$

Gibbs sampling "steps"

1. Start from initial arbitrary values of the parameters
   - θ^0: $\theta_1^0, \theta_2^0, \ldots, \theta_K^0$
2. Draw samples:

$$\theta_1^1 \sim f(\theta_1^1 | \theta_2^0, , \theta_K^0, \text{Data})$$
$$\theta_2^1 \sim f(\theta_2 | \theta_1^1, \theta_3^0, \ldots, \theta_K^0, \text{Data})$$
$$\ldots$$
$$\theta_K^1 \sim f(\theta_K | \theta_1^1, \theta_2^1, \ldots, \theta_{K-1}^1, \text{Data})$$

3. Repeat step 2 S times, obtaining values

$$\theta_k^s \sim f(\theta_k | \theta_1^s, \ldots \theta_{k-1}^s, \ldots, \theta_K^{s-1}, \text{Data})$$

   at each iteration $s = 2, \ldots, S$ of the algorithm

Gibbs sampling in practice - Example

- Let $Y_1, Y_2, \ldots, Y_n$ be a random sample from $N(\mu, \sigma^2)$
  - μ, σ^2 unknown
- Goal: "estimate" μ y σ^2 using Bayesian inference (MCMC simulations)
  - i.e., derive the **posterior distributions** of μ and σ^2
- We resort to the usual "Bayesian mantra":
  - **Posterior distribution** $\propto$ Likelihood $\times$ **Prior distribution**
- In this example:

$$f(\mu, \sigma^2|\mathrm{Y}) \propto f(\mathrm{Y}|\mu, \sigma^2) \times f(\mu, \sigma^2) \tag{18}$$

- Let's follow the 4 steps of Bayesian inference seen before:

  1. Defining the likelihood for $(Y_1, Y_2, \ldots, Y_n)$, given μ y σ^2
  2. Specifying the **prior distributions** for parámeters μ and σ
  3. Deriving - or approximating - the **posterior distributions** of μ and θ
  4. Summarizing these **posterior distributions**

Step 1: specifying $f(\mathrm{Y}|\mu, \sigma^2)$

- $f(\mathrm{Y}|\mu, \sigma^2)$ is given by:

$$f(\mathrm{Y}|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \\ \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \tag{19}$$

- Hence, the posterior distribution for $f(\mu, \sigma^2|\mathrm{Y})$ will be given by:

$$f(\mu, \sigma^2|\mathrm{Y}) \propto \frac{1}{\sigma^{2n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \; f(\mu, \sigma^2) \tag{20}$$

Step 2: Prior distributions for $f(\mu, \sigma^2)$

- Two options:
  1. Defining a **joint prior** distribution for μ and σ^2, $f(\mu, \sigma^2)$
  2. Defining *a priori* independent **prior distributions** for μ and σ^2
     - i.e., assuming that the *prior* distribution about μ is independent from the *prior* distribution about σ^2

$$f(\mu, \sigma^2) = f(\mu) \times f(\sigma^2) \tag{21}$$

- The second approach is generally easier

- How do we choose $f(\mu)$ y $f(\sigma^2)$?
- **Conjugate priors** are always useful when applying Gibbs sampling
  - because this will help obtaining (**conditional**) posterior distributions for μ and σ^2 from which it is easy to sample
  - property of **conjugate** prior distributions

- How do we choose $f(\mu)$ y $f(\sigma^2)$?
- **Conjugate priors** are always useful when applying Gibbs sampling
  - because this will help obtaining (**conditional**) posterior distributions for μ and σ^2 from which it is easy to sample
  - property of **conjugate** prior distributions
- The **conjugate** prior for μ is the Normal distribution:

$$f(\mu) \propto N(m, \sigma_\mu^2) \tag{22}$$

  where m and σ_μ^2 are **hyperparameters** of this **prior distribution**

- Easy to simulate from, using "rnorm"

- If the analyst has *prior* information about μ
  - e.g., from previous studies

  she can use this information to specify m and σ_μ^2
  - $\Rightarrow$ the **prior distribution** for μ would then be both **conjugate** and **informative**

- If the analyst has *prior* information about μ
  - e.g., from previous studies

  she can use this information to specify m and σ_μ^2
  - $\Rightarrow$ the **prior distribution** for μ would then be both **conjugate** and **informative**

- If there is no prior information about μ, the analyst would go for a **conjugate** and **vague** prior distribution
- For instance:
  - $\mu = 0$
  - σ_μ^2 "large" (e.g., 100)

- The **conjugate prior** distribution for the variance of a normal variable is the **Inverse Gamma**
- The pdf of an Inverse Gamma distribution is

$$f(\sigma^2) \propto (\sigma^2)^{-(\alpha+1)} exp(-\beta/\sigma^2) \tag{23}$$

with **hyperparameters**:

  - $\alpha > 0$, affecting the shape of $f(\sigma^2)$
  - $\beta > 0$, affecting the scale of $f(\sigma^2)$

- Note that R does not have a "canned command" to generate values from an Inverse Gamma distribution

- We would need to use specialized packages (like "bayesAB")
- Or - even easier - resort to the following property:
  - If $\sigma^2 \sim$ Gamma Inversa(α, β)
  - $\Rightarrow 1/\sigma^2 \sim$ Gamma(α, β)
- We can use the "rgamma" command in R to generate $1/\sigma^2$
  - and then obtain the inverse

Step 3: Deriving **posterior distributions** for μ and σ^2

- From (20), (22) and (23), we have that:

$$f(\mathrm{Y}|\mu,\sigma^2) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right)$$

$$f(\mu) \propto \frac{1}{(\sigma_\mu^2)^{1/2}} exp\left(\frac{-(\mu-m)^2}{2\sigma_\mu^2}\right)$$

$$f(\sigma^2) \propto \frac{1}{(\sigma^2)^{(\alpha+1)}} exp(-\beta/\sigma^2)$$

- Hence, the **posterior distribution** of the parameters is:

$$
\begin{aligned}
f(\mu, \sigma^2|\mathrm{Y}) \propto & \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right) \times \\
& \frac{1}{(\sigma_\mu^2)^{1/2}} exp\left(-\frac{(\mu-m)^2}{2\sigma_\mu^2}\right) \times \\
& \frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2)
\end{aligned}
\tag{24}
$$

- Problem: (24) has no "known" form
  - it is difficult to obtain means, variances, etc. analytically
  - but also impossible to simulate from (24)!

- What can we do?

- What can we do? $\Rightarrow$ Gibbs sampling!

- What can we do? $\Rightarrow$ Gibbs sampling!
- We cannot draw samples from $f(\mu, \sigma^2|\mathrm{Y})$
- But let's try to express $f(\mu, \sigma^2|\mathrm{Y})$ as:

$$f(\mu, \sigma^2|\mathrm{Y}) = f(\mu|\sigma^2, \mathrm{Y}) \times f(\sigma^2|\mu, Y) \tag{25}$$

where:

- $f(\mu|\sigma^2, \mathrm{Y})$ is the **conditional posterior** distribution of μ
  - taking σ^2 (and obviously Y) as given
- $f(\sigma^2|\mu, \mathrm{Y})$ is the **conditional posterior** distribution of σ^2
  - taking μ (and Y) as given

Deriving $f(\mu|\sigma^2, Y)$

- From $f(\mu, \sigma^2|Y)$:

$$f(\mu, \sigma^2|Y) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times$$
$$\frac{1}{(\sigma_\mu^2)^{1/2}} exp\left(-\frac{(\mu - m)^2}{2\sigma_\mu^2}\right) \times$$
$$\frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2)$$

- We take all parameters other than μ as constant, and focus only on the terms that depend on μ

Deriving $f(\mu|\sigma^2, Y)$

- From $f(\mu, \sigma^2|Y)$:

$$f(\mu, \sigma^2|Y) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times \frac{1}{(\sigma_\mu^2)^{1/2}} exp\left(-\frac{(\mu - m)^2}{2\sigma_\mu^2}\right) \times \frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2)$$

- We take all parameters other than μ as constant, and focus only on the terms that depend on μ

$$f(\mu|\sigma^2, Y) \propto exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times exp\left(-\frac{(\mu - m)^2}{2\sigma_\mu^2}\right)$$

- With a little bit of algebra:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2} - \frac{(\mu-m)^2}{2\sigma_\mu^2}\right)$$

- With a little bit of algebra:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2} - \frac{(\mu-m)^2}{2\sigma_\mu^2}\right)$$

$$f(\mu|\sigma^2, Y) \propto exp\left(-\frac{\sum_{i=1}^{n} y_i^2 - 2\mu\sum_{i=1}^{n} y_i + \sum_{i=1}^{n}\mu^2}{2\sigma^2} - \frac{\mu^2 - 2\mu m + m^2}{2\sigma_\mu^2}\right)$$

- With a little bit of algebra:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}-\frac{(\mu-m)^2}{2\sigma_\mu^2}\right)$$

$$f(\mu|\sigma^2, Y) \propto exp\Bigg(-\frac{\sum_{i=1}^{n} y_i^2 - 2\mu\sum_{i=1}^{n} y_i + \sum_{i=1}^{n}\mu^2}{2\sigma^2} - \frac{\mu^2 - 2\mu m + m^2}{2\sigma_\mu^2}\Bigg)$$

$$exp\Big(-\frac{1}{2}\Big[\frac{\sigma_\mu^2\sum_{i=1}^{n} y_i^2 - 2\sigma_\mu^2\,\mu\, n\,\bar{y} + n\,\sigma_\mu^2\,\mu^2 + \sigma^2\,\mu - 2\,\sigma^2\,\mu\, m + \sigma^2\, m}{\sigma^2\sigma_\mu^2}\Big]\Big)$$

- And grouping all the terms that depend on μ:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left(-\frac{1}{2}\left[\frac{\mu^2(n\sigma_\mu^2 + \sigma^2) - 2\mu(\sigma_\mu^2 n\bar{y} + \sigma^2 m) + c}{\sigma^2\sigma_\mu^2}\right]\right)$$

where $c = \sigma_\mu^2 \sum_{i=1}^{n} y_i^2 + \sigma^2 m$ is a constant (with respect to μ)

$$\Rightarrow f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left(-\frac{1}{2}\left[\frac{\mu^2(n\sigma_\mu^2 + \sigma^2) - 2\mu(\sigma_\mu^2 n\bar{y} + \sigma^2 m)}{\sigma^2\sigma_\mu^2}\right]\right)$$
$$exp\left(-\frac{1}{2}\frac{c}{\sigma^2\sigma_\mu^2}\right)$$

$$f(\mu|\sigma^2, \mathsf{Y}) \propto exp\left(-\frac{1}{2}\left[\frac{\mu^2(n\sigma_\mu^2 + \sigma^2) - 2\mu(\sigma_\mu^2 n\bar{y} + \sigma^2 m)}{\sigma^2\sigma_\mu^2}\right]\right)$$

$$f(\mu|\sigma^2, \mathbf{Y}) \propto exp\left(-\frac{1}{2}\left[\frac{\mu^2(n\sigma_\mu^2 + \sigma^2) - 2\mu(\sigma_\mu^2 n\bar{y} + \sigma^2 m)}{\sigma^2\sigma_\mu^2}\right]\right)$$

- Dividing the numerator and denominator by $n\sigma_\mu^2 + \sigma^2$:

$$f(\mu|\sigma^2, \mathbf{Y}) \propto exp\left(-\frac{1}{2}\left[\frac{\mu^2 - 2\mu\frac{(\sigma_\mu^2 n\bar{y} + \sigma^2 m)}{(n\sigma_\mu^2 + \sigma^2)}}{\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2 + \sigma^2}}\right]\right) \qquad (26)$$

- (26) now resembles the kernel of a known distribution ...

- Completing the square:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left[-\frac{\left(\mu - \frac{\sigma_\mu^2 n\bar{y}+\sigma^2 m}{n\sigma_\mu^2+\sigma^2}\right)^2}{2\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2+\sigma^2}}\right] \times exp\left[-\frac{\left(\frac{\sigma_\mu^2 n\bar{y}+\sigma^2 m}{n\sigma_\mu^2+\sigma^2}\right)^2}{2\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2+\sigma^2}}\right]$$

- And since the second term is again constant (with respect to μ):

$$f(\mu|\sigma^2, \mathrm{Y}) \propto exp\left[-\frac{\left(\mu - \frac{\sigma_\mu^2 n\bar{y}+\sigma^2 m}{n\sigma_\mu^2+\sigma^2}\right)^2}{2\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2+\sigma^2}}\right] \tag{27}$$

- (27) shows that $f(\mu|\sigma^2, \mathrm{Y})$ is Normal, with mean:

$$\frac{\sigma_\mu^2 n\bar{y} + \sigma^2 m}{n\sigma_\mu^2 + \sigma^2} \tag{28}$$

- and variance:

$$\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2 + \sigma^2} \tag{29}$$

- The important part is: **we know** how to draw samples from (27)
  - e.g., using the "rnorm" command in R

- Note that the expectation of $f(\mu|\sigma^2, \mathrm{Y})$ is a combination of:
  - the **prior mean** m
  - and the sample mean, $\bar{y}$
- The relative weight of m and $\bar{y}$ depends on:
  1. the sample size, n: the larger the value of n, the "heavier" the weight of the data vis-á-vis m
  2. the prior variance, σ^2_μ: the larger the value of σ^2_μ, the heavier the weight of the data vis-a-vis the prior information
     - when there is more uncertainty about the prior information, we place more emphasis on the data
  3. the variance σ^2: the larger the value of σ^2, the heavier the weight of m
     - when the data is less informative, we place more emphasis on m

Derivation of $f(\sigma^2|\mu, \mathsf{Y})$

- To derive $f(\sigma^2|\mu, \mathsf{Y})$, we start again from $f(\mu, \sigma^2|\mathsf{Y})$ and focus only on those terms that depend on σ^2
  - treating every other term as "constant"

Derivation of $f(\sigma^2|\mu, \mathrm{Y})$

- To derive $f(\sigma^2|\mu, \mathrm{Y})$, we start again from $f(\mu, \sigma^2|\mathrm{Y})$ and focus only on those terms that depend on σ^2
  - treating every other term as "constant"
- That is, starting from:

$$f(\mu, \sigma^2|\mathrm{Y}) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times \frac{1}{(\sigma_\mu^2)^{1/2}} exp\left(-\frac{(\mu - m)^2}{2\sigma_\mu^2}\right) \times \frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2)$$

$$f(\sigma^2|\mu, Y) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2) \tag{30}$$

- And, grouping terms:

$$f(\sigma^2|\mu, Y) \propto \frac{1}{(\sigma^2)^{(n/2+\alpha+1)}} exp\left[-\left(\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2} + \frac{\beta}{\sigma^2}\right)\right]$$

$$f(\sigma^2|\mu, Y) \propto \frac{1}{(\sigma^2)^{n/2}} exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma^{2(\alpha+1)}} exp(-\beta/\sigma^2) \tag{30}$$

- And, grouping terms:

$$f(\sigma^2|\mu, Y) \propto \frac{1}{(\sigma^2)^{(n/2+\alpha+1)}} exp\left[-\left(\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2} + \frac{\beta}{\sigma^2}\right)\right]$$

$$f(\sigma^2|\mu, Y) \propto \frac{1}{(\sigma^2)^{(n/2+\alpha+1)}} exp\left(-\frac{\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2} + \beta}{\sigma^2}\right) \tag{31}$$

- (31) is the kernel of an Inverse Gamma distribution, with parameters:
  - $\alpha^{'} = n/2 + \alpha$
  - $\beta^{'} = \frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2} + \beta$
- And, again, we know how to draw samples from this distribution!

Recap - Gibbs sampling in this example

- In sum, we had arrived to a **joint posterior distribution** we did not know how to sample from: $f(\mu, \sigma^2|\mathrm{Y})$
- But we could "split" this joint posterior in two **conditional posterior distributions**, from which it is easy to draw samples
  1. A normal distribution, $f(\mu|\sigma^2, \mathrm{Y})$
  2. And an inverse gaussian distribution, $f(\sigma^2|\mu, \mathrm{Y})$
- Hence, rather than drawing samples of μ and σ from $f(\mu, \sigma^2|\mathrm{Y})$, we obtain them from (1) and (2)

Steps in our Gibbs sampling algorithm

1. Choose the hyperparameters for the **prior distributions** for μ y σ^2
   - m and σ^2_μ
   - α and β
2. Propose initial values for μ_0 y σ^2_0
3. Draw σ^2_1 from an Inverse Gamma distribution with parameters

$$\alpha' = n/2 + \alpha$$

$$\beta' = \frac{\sum_{i=1}^{n}(y_i - \mu_0)^2}{2} + \beta$$

4. Draw μ_1 from a Normal distribution with:

$$\text{Mean} = \frac{\sigma_\mu^2 n\bar{y} + \sigma_1^2 m}{n\sigma_\mu^2 + \sigma_1^2}$$

$$\text{Variance} = \frac{\sigma_1^2 \sigma_\mu^2}{n\sigma_\mu^2 + \sigma_1^2}$$

5. Repeat steps 3 and 4 for $s = 2, \ldots, S$, taking μ_{s-1} and σ_{s-1}^2 as starting values for iteration s

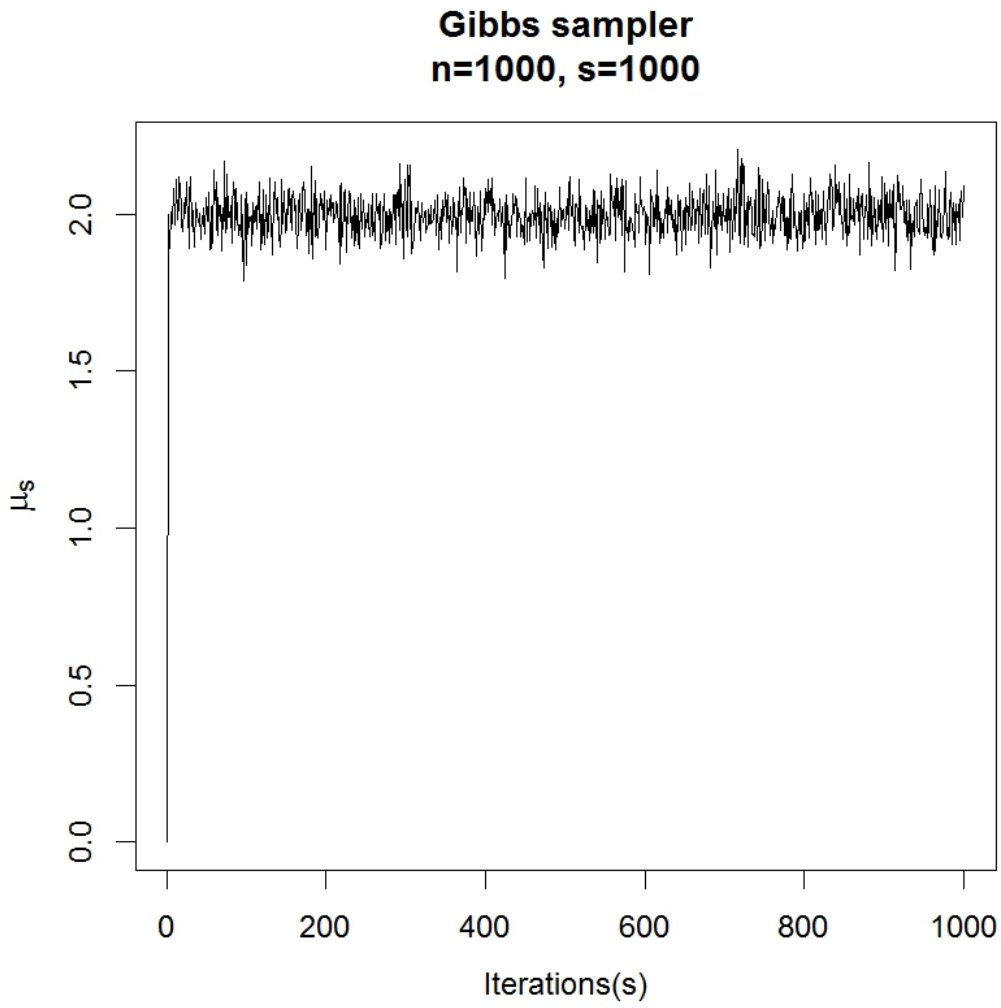
   - S must be large enough so that convergence to the **stationary distribution** occurs
   - i.e., large enough for the sample draws to their "stationary state"

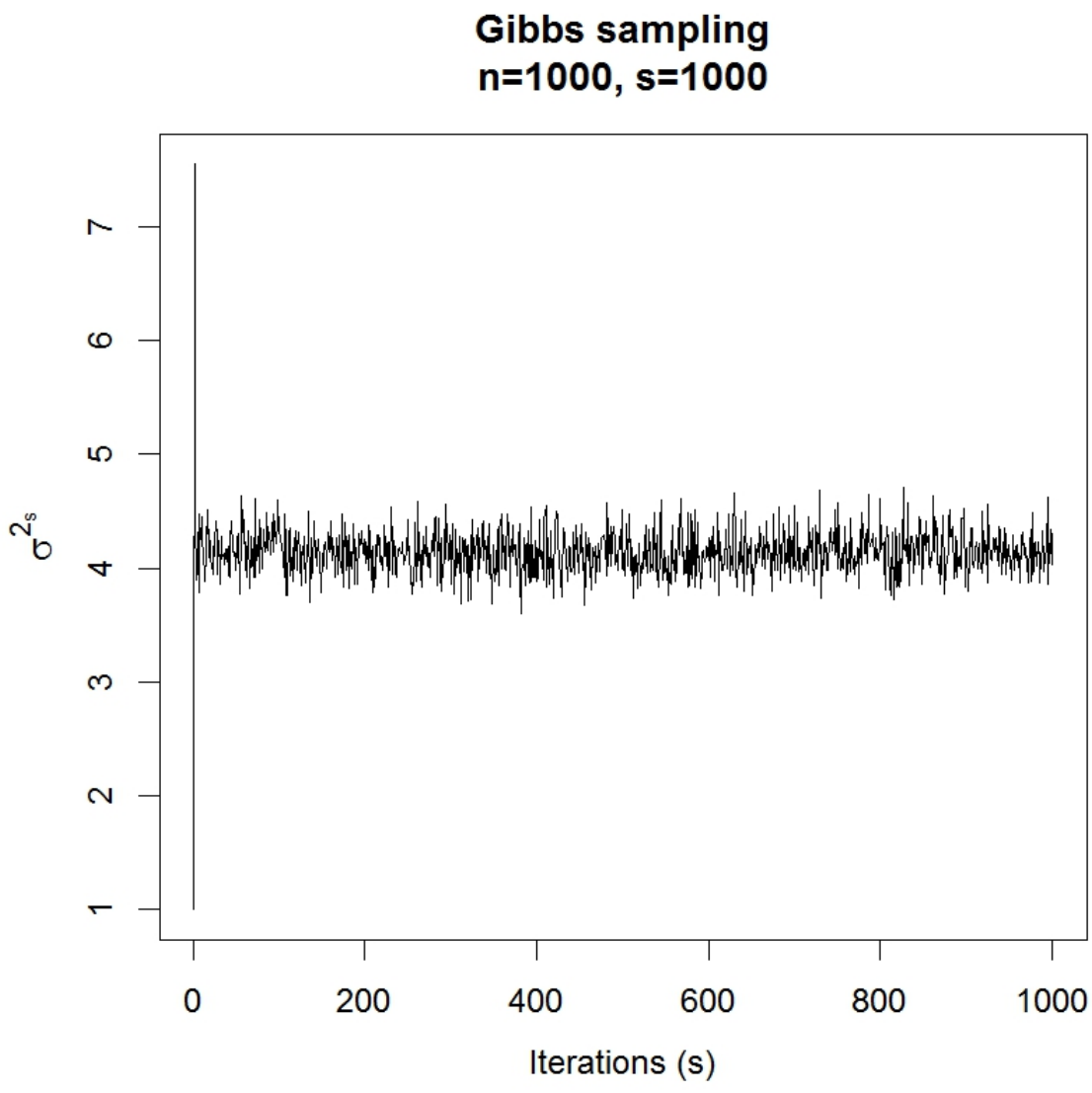
Gibbls sampling in R

- R: “Gibbs sampling - Normal distribution.R”

Gibbls sampling in R

- R: “Gibbs sampling - Normal distribution.R”




Gibbs sampling
n=1000, s=1000
σ^{2_s}
1
2
3
4
5
6
7
0
200
400
600
800
1000
Iterations (s)

Exercise - Gibbs sampling

- Assume we have the same data
  - i.e., a sample $Y_1, Y_2, \ldots, Y_n$ from a $N(\mu, \sigma^2)$
- But suppose we adopt the following **prior distributions** for the parameters:

$$f(\mu) = constant$$
$$f(\sigma^2) \propto 1/\sigma^2$$

- The exercise consists of 2 tasks

1. Verify that the resulting **conditional posterior distributions** are:

$$f(\mu|\sigma^2, \mathrm{Y}) \propto N\left(\bar{y}, \frac{\sigma^2}{n}\right)$$

$$f(\sigma^2|\mu, \mathrm{Y}) \propto \text{Gamma Inversa}(\alpha, \beta)$$

with:

$$\alpha = n/2$$

$$\beta = \frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2}$$

2. Verify that you understand the new sampler, contained in "Gibbs sampling - Normal distribution, alternative priors.R"

Metropolis-Hastings (MH) algorithm

- For some models, drawing samples from the **conditional posterior distributions** may be unfeasible
- Hence, Gibbs sampling is not applicable
- However, the M-H algorithm can be used

Metropolis-Hastings (MH) algorithm

- For some models, drawing samples from the **conditional posterior distributions** may be unfeasible
- Hence, Gibbs sampling is not applicable
- However, the M-H algorithm can be used
- Specifically, suppose it is not possible to draw samples of Θ from its **joint posterior density**
    - or from any of θ's **conditional posterior** densities
- However, assume we can draw samples from another distribution, $g()$
- Then, we can apply the M-H algorithm

MH algorithm in a nutshell

1. Start with an initial value of θ, say θ^0
2. Draw a "candidate" value for θ, θ^c, from a proposal density $g()$, conditional on θ^{s-1}, $s = 1, \ldots, S$
   - i.e., $\theta^c \sim g(\theta|\theta^{s-1})$ (Markov chain)
3. Compute:

$$Ratio = \frac{f(\theta^c|\text{Data}) \times g(\theta^{s-1}|\theta^c)}{f(\theta^{s-1}|\text{Data}) \times g(\theta^c|\theta^{s-1})} \tag{32}$$

4. Generate $u \sim \text{Uniform}(0, 1)$

5. Compare u against the Ratio in (32):
    - If Ratio $> u$, $\theta^s = \theta^c$
    - If Ratio $\leq u$, $\theta^s = \theta^{s-1}$
6. Go back to step 2

Some comments about the MH algorithm

- In step 2, the value of θ drawn from $g()$ is initially a "candidate", because it is not immediately accepted
  - this depends on the comparison between u and the Ratio in (32)

Some comments about the MH algorithm

- In step 2, the value of θ drawn from $g()$ is initially a "candidate", because it is not immediately accepted
  - this depends on the comparison between u and the Ratio in (32)
- Note that this Ratio has two "components":
  1. the relationship between the posterior distributions evaluated at θ^c y θ^{s-1}

$$\frac{f(\theta^c|\text{Data})}{f(\theta^{s-1}|\text{Data})}$$

- The larger this ratio, the higher the likelihood that θ^c is a "good candidate"

2. The second component is

$$\frac{g(\theta^{s-1}|\theta^{c})}{g(\theta^{c}|\theta^{s-1})}$$

the relationship between the proposal densities evaluated at the previous value of the parameter θ and at the "candidate" value

- This can be understood as an "adjustment" to take into consideration how likely is θ^{c} with respect to θ^{s}
- That is, we are adjusting for the "quality" of the proposal density
  - some $g()s$ may disproportinally "choose"' very likely/unlikely values of θ^{c}

Some commonly used proposal densities $g()$

- Normal random-walk:

$$\theta^c \sim N(\theta^{s-1}, \sigma_c^2)$$

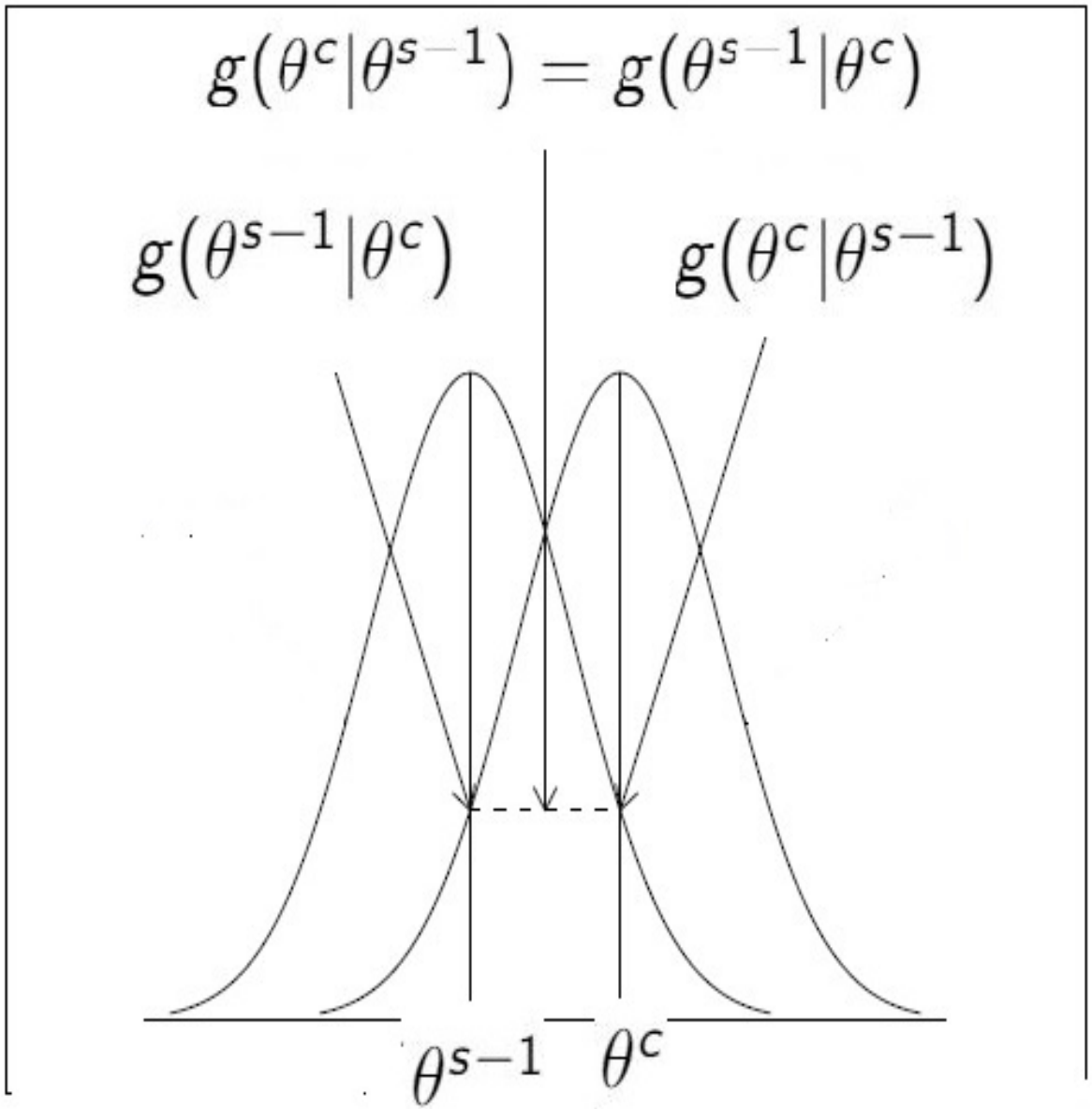
with σ_c^2 "large"

- Uniform random-walk:

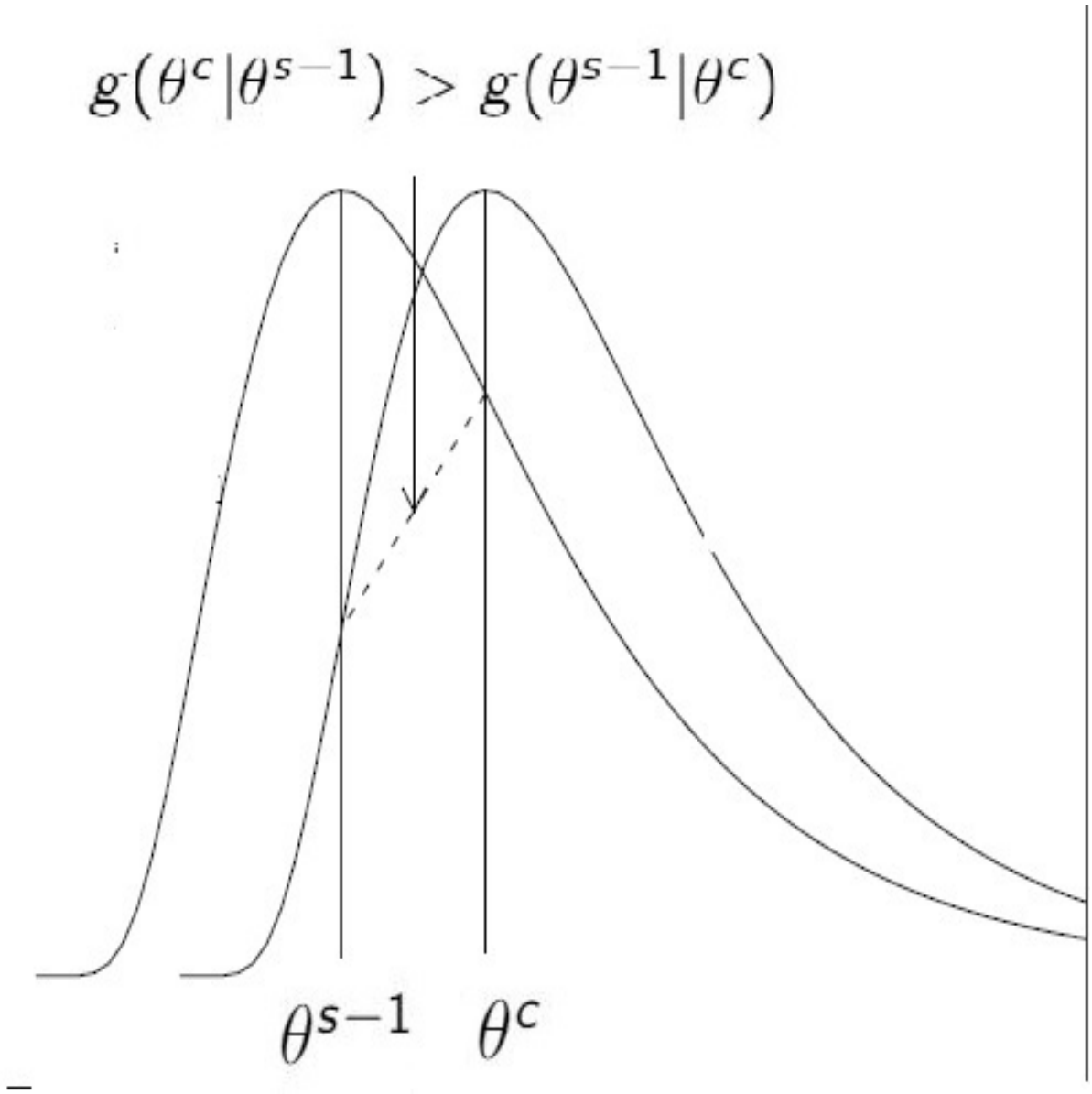
$$\theta^c \propto \theta^{s-1} + U(a, b)$$

- The key advantage of these proposals is that they are symmetric:
  - $g(\theta^c|\theta^{s-1}) = g(\theta^{s-1}|\theta^c)$

A symmetric proposal



An asymmetric proposal



- With symmetric proposal, the Ratio in (32) becomes simpler:

$$Ratio = \frac{f(\theta^c|\text{Data})}{f(\theta^{s-1}|\text{Data})} \tag{33}$$

- In general, we will work with the logarithm of this ratio:

$$log(Ratio) = log(f(\theta^c|\text{Data})) - log(f(\theta^{s-1}|\text{Data}) \tag{34}$$

and compare it against $log(u)$, $u \sim \text{Uniform}(0, 1)$

- For the same reason that we work with the log-likelihood function in ML estimation
  - stability, avoiding overflows

Example: "Estimating" the correlation coefficient of a bivariate Normal distribution using the MH algorithm

- Assume that $(x, y) \sim$ Bivariate Normal:
  - $\mu_x = \mu_y = 0$
  - $\sigma_x^2 = \sigma_y^2 = 1$

  with density function:

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2)}} exp\left(\frac{x^2 - 2\rho xy - y^2}{2(1-\rho^2)}\right) \tag{35}$$

- Suppose we have a sample (X_i, Y_i), $i = 1, \ldots, n$
- Goal: draw inferences about ρ

- As always:

$$\textbf{Posterior distribution} \propto \textbf{Likelihood} \times \textbf{Prior}$$

$$f(\rho|x, y) \propto f(x, y|\rho) \times f(\rho)$$

- Let's assume we have no *prior* information abut ρ
  - except that $\rho \in (-1, 1)$
- **Vague prior distribution** for ρ: Uniform$(-1, 1)$

$$f(\rho) = 1/2$$

- The **posterior distribution** for ρ is then:

$$f(\rho|x,y) \propto \frac{1}{(1-\rho^2)^{n/2}} exp\left(\frac{\sum_{i=1}^{n} x_i^2 - 2\rho\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{2} y_i^2}{2(1-\rho^2)}\right)$$

- No closed form
  - cannot draw samples from $f(\rho|x,y)$
- Can we use Gibbs sampling to draw from the **conditional posterior**?
  - No (why?)
- $\Rightarrow$ we are going to resort to the MH algorithm

- To apply MH algorithm, we need to choose a suitable proposal density $g()$
- For instance, we can use a symmetric proposal
  - because - as we saw - this simplifies computations
- A possible proposal:

$$\rho^c = \rho^{s-1} + \text{Uniform}(a, b) \tag{36}$$

  with $-1 < a < 0$, $0 < b < 1$ chosen so as to allow the algorithm to "explore" the parameter space
- For instance:

$$\rho^c = \rho^{s-1} + \text{Uniform}(-0.1, 0.1)$$

Steps of our MH algorithm

1. Start from an initial value of ρ
   - e.g., $\rho^0 = 0$
2. For $s = 1, \ldots, S$, draw a "candidate" ρ:

$$\rho^c = \rho^{s-1} + \text{Uniform}(-0.1, 0.1)$$

3. Evaluate $f(\rho|x, y)$ at:
   - ρ^c: $log(f(\rho^c|x, y))$
   - ρ^s: $log(f(\rho^{s-1}|x, y))$

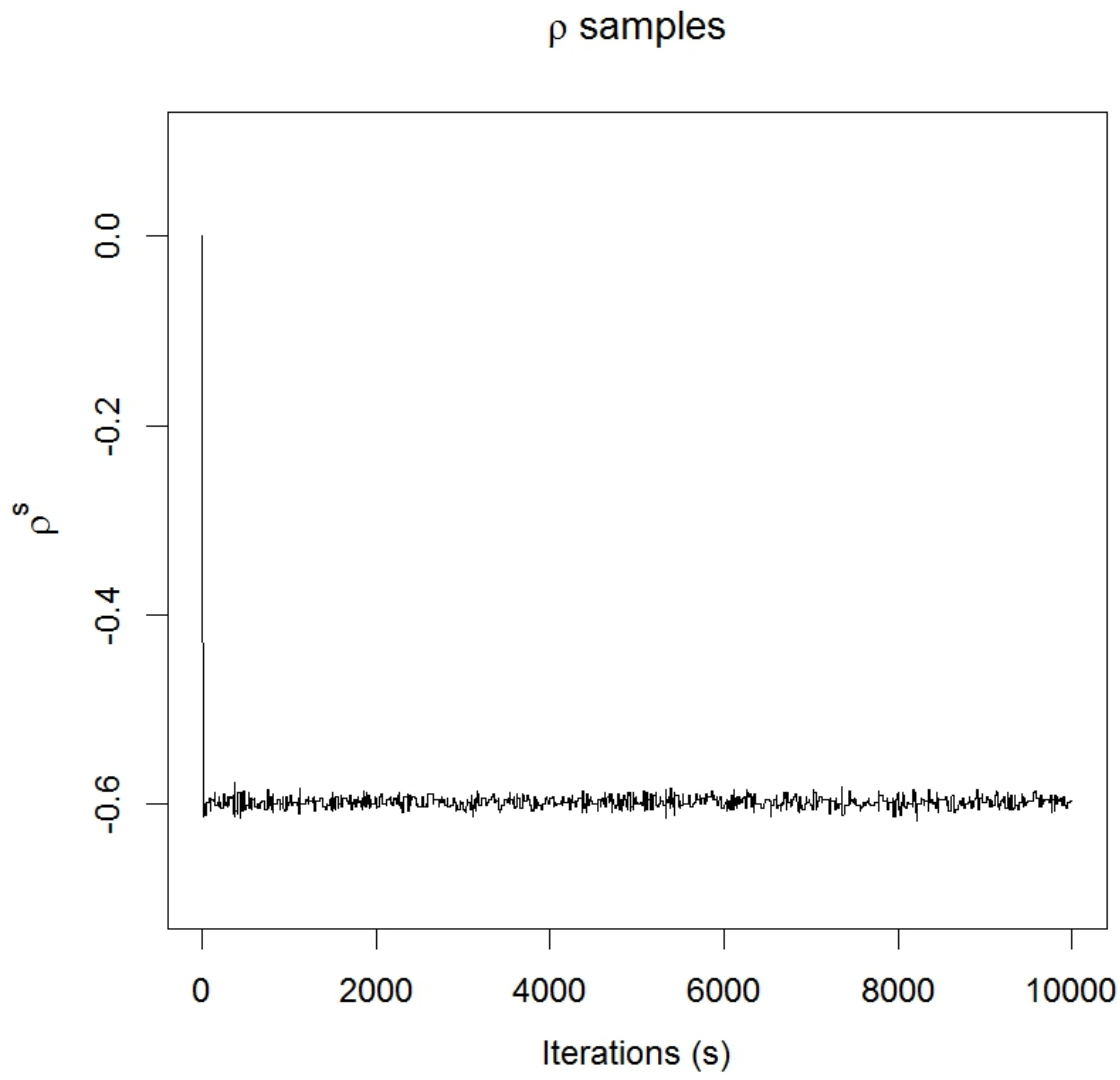
   and compute the log-ratio: $log(f(\rho^c|x, y) - log(f(\rho^{s-1}|x, y)$

4. Draw $u \sim \text{Uniform}(0, 1)$
5. Compare the log-ratio (step 3) against $log(u)$
    - If $(log(f(\rho^c|x, y) - log(f(\rho^{s-1}|x, y)) > log(u) \Rightarrow \rho^s = \rho^c$
    - Else if $(log(f(\rho^c|x, y) - log(f(\rho^{s-1}|x, y)) \leq log(u) \Rightarrow \rho^s = \rho^{s-1}$

4. Draw $u \sim \text{Uniform}(0, 1)$
5. Compare the log-ratio (step 3) against $log(u)$
    - If $(log(f(\rho^c|x, y) - log(f(\rho^{s-1}|x, y)) > log(u) \Rightarrow \rho^s = \rho^c$
    - Else if $(log(f(\rho^c|x, y) - log(f(\rho^{s-1}|x, y)) \leq log(u) \Rightarrow \rho^s = \rho^{s-1}$

- R: "MH algorithm for rho.R"

MH algorithm - sampled values for ρ



Combining Gibbs sampler & Metropolis steps

- For most social science models, we are typically going to use a combination of Gibbs sampling and Metropolis steps models
- Example: hierarchical logit model:
  - i individual-level observations, $i = 1, \ldots N$
  - in $j = 1, \ldots J$ countries

$$P(Y_{i,j} = 1) = \frac{exp(X'_{i,j}\beta + \eta_j)}{1 + exp(X'_{i,j}\beta + \eta_j)} \tag{37}$$

with $\eta_j \sim N(0, \sigma^2)$

- Parameters:
  1. β
  2. η_j, $j = 1, \ldots, J$
  3. σ^2

- Assuming:
  1. $N(0, 100I)$ **priors** for β
  2. $N(0, 100)$ **priors** for η_j, $j = 1, \ldots, J$
  3. And **conjugate** Inverse Gamma (0.1, 0.1) priors for σ^2

  we can show that:

  - β and η_j have no closed-form **conditional posterior distributions**
  - but σ^2 has an Inverse Gamma **conditional posterior distribution**

- Specifically, the **conditional posterior distributions** for the parameters are:

$$f(\beta|\eta,\sigma^2) \propto \frac{exp(X'_{i,j}\beta+\eta_j)}{1+exp(X'_{i,j}\beta+\eta_j)} \times N(0,100I) \tag{38}$$

$$f(\eta_j|\beta,\eta_{-j},\sigma^2) \propto \frac{exp(X'_{i,j}\beta+\eta_j)}{1+exp(X'_{i,j}\beta+\eta_j)} \times N(0,100) \tag{39}$$

$$f(\sigma^2|\beta,\eta) \propto \frac{1}{(\sigma^2)^{(J/2+0.1+1)}} exp\left(-\frac{0.1+\frac{\sum_{j=1}^{J}\eta_j^2}{2}}{\sigma^2}\right) \tag{40}$$

- The distributions in (38) and (39) have no closed form
- But (40) is the kernel of an Inverse Gamma distribution with parameters $0.1 + J/2$ and $0.1 + \frac{\sum_{j=1}^{2} \eta_j^2}{2}$
- Hence, we will need to resort to
  1. M-H steps to draw samples for β and η_j, $j = 1, \ldots, J$
  2. Gibbs sampling to draw samples for σ^2
- Application: "Hierarchical Logit.R"

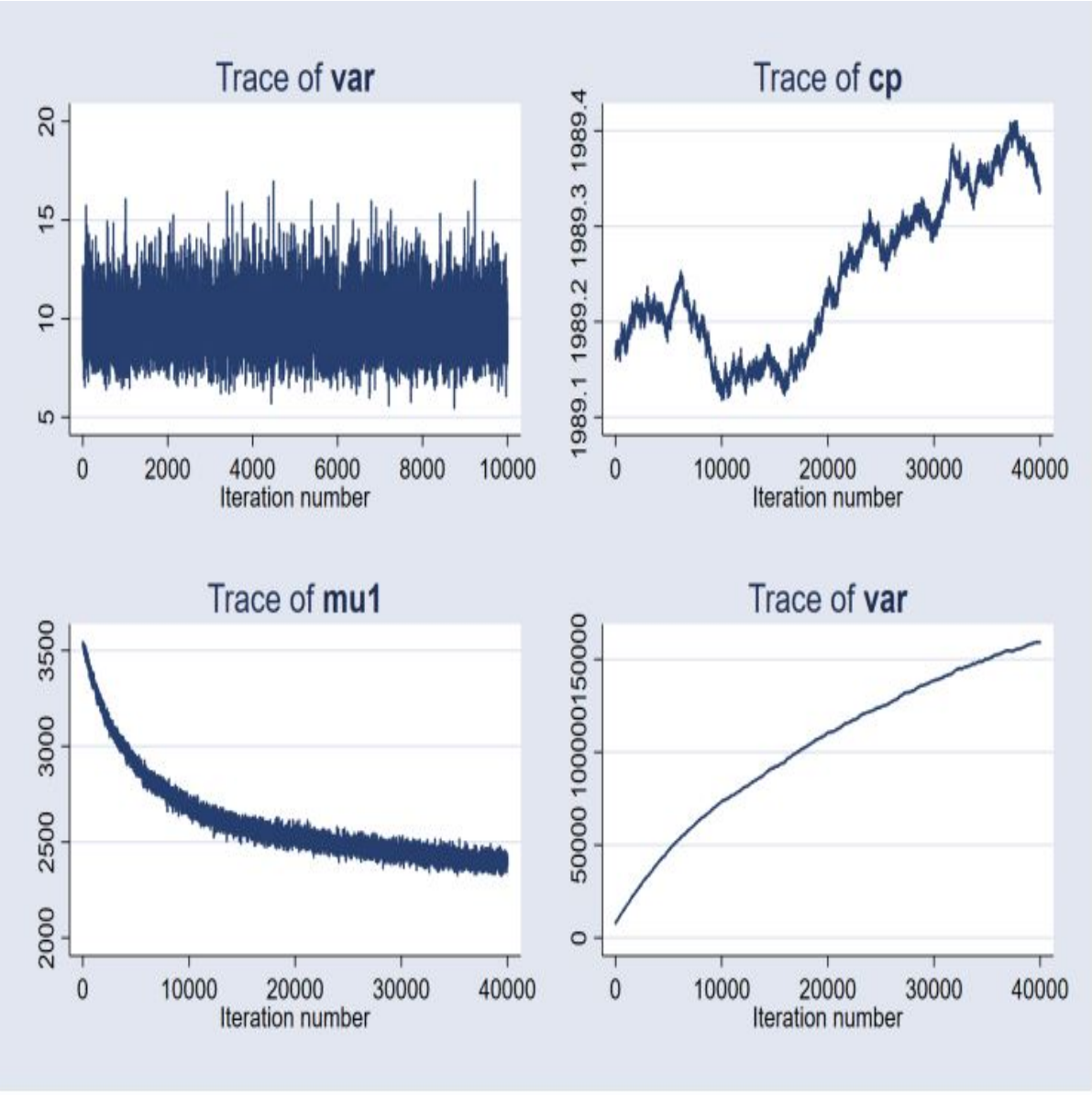
Exercise

- If we use a probit rather than a logit model, all the **conditional posterior distributions** have known closed forms
- So, when fitting a hierarchical probit model, we only need Gibbs sampling
    - no M-H steps are needed
- Convergence is typically faster.
- Application: "Hierarchical Probit.R"

Assessing Convergence

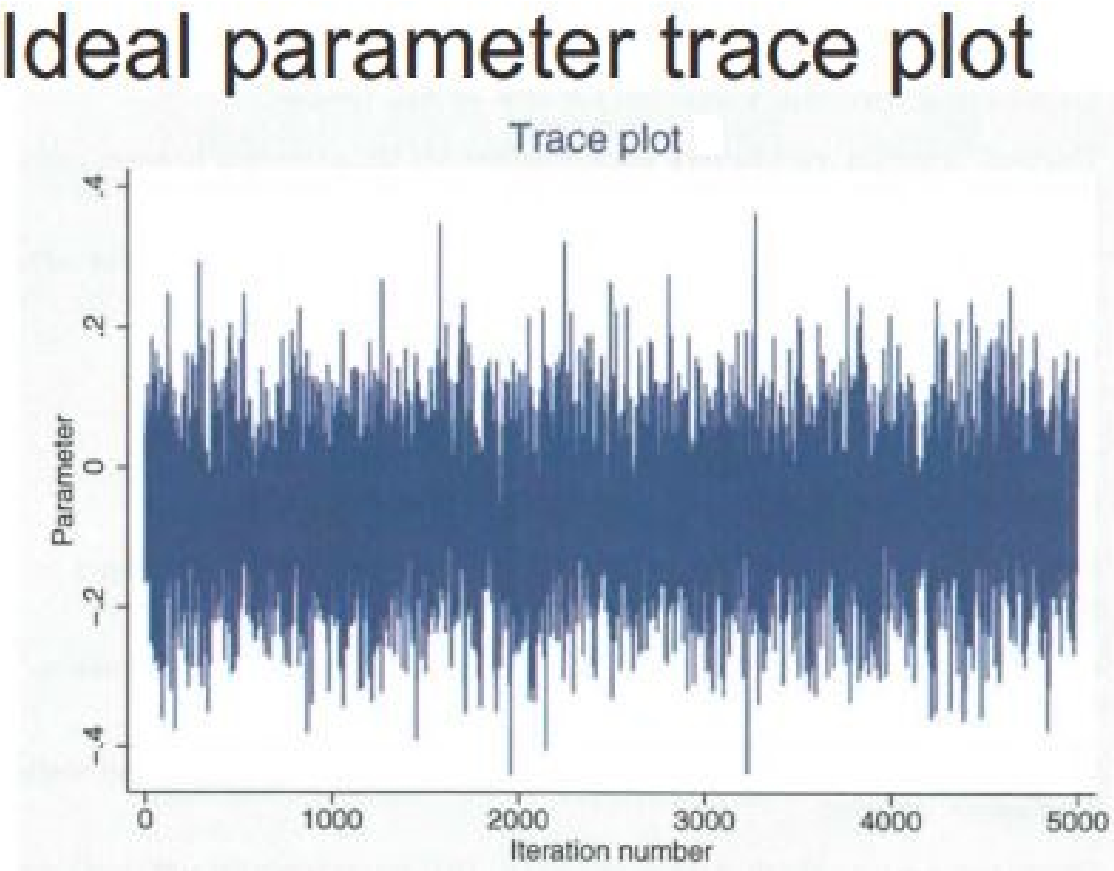
- In our exercises - Gibbs sampling, MH algorithm - we run the sampler for an "arbitrarily long" number of iterations
- And visually explored the traceplots
  - i.e., checked that the sampled values seemed to reach a "stable state"
- Visual inspection of trace plots is a first - informal - approach to assesing convergence of the MCMC algorithm

Traceplots for different parameter draws



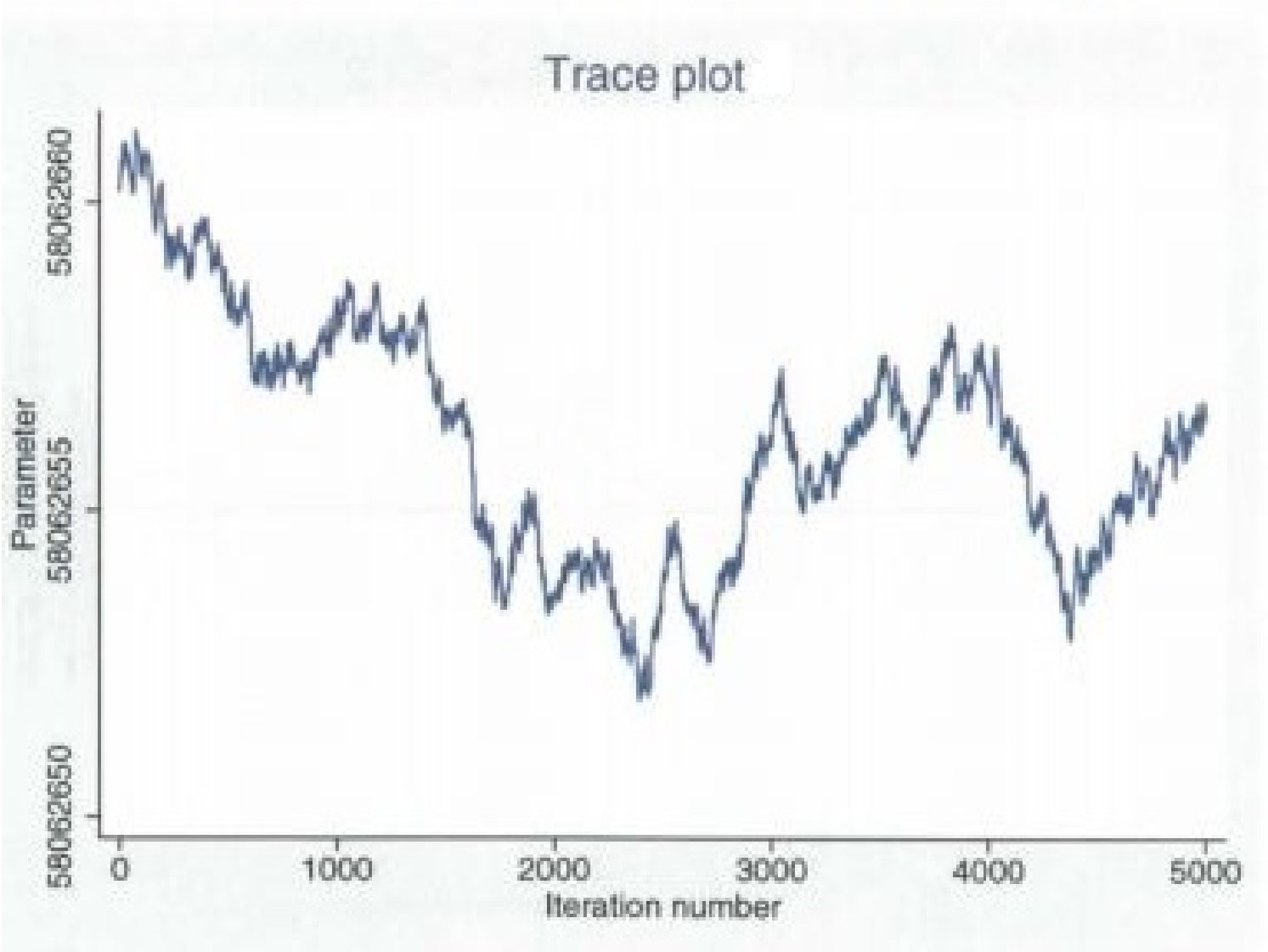
Informal approach to assessing convergence - checking traceplots

- For a well-mixing, convergent parameter, simulated values look almost vertical and dense
- stable values after burn-in (no trends)



- Not like this:

MCMC did not converge



Formal approaches to assessing convergence

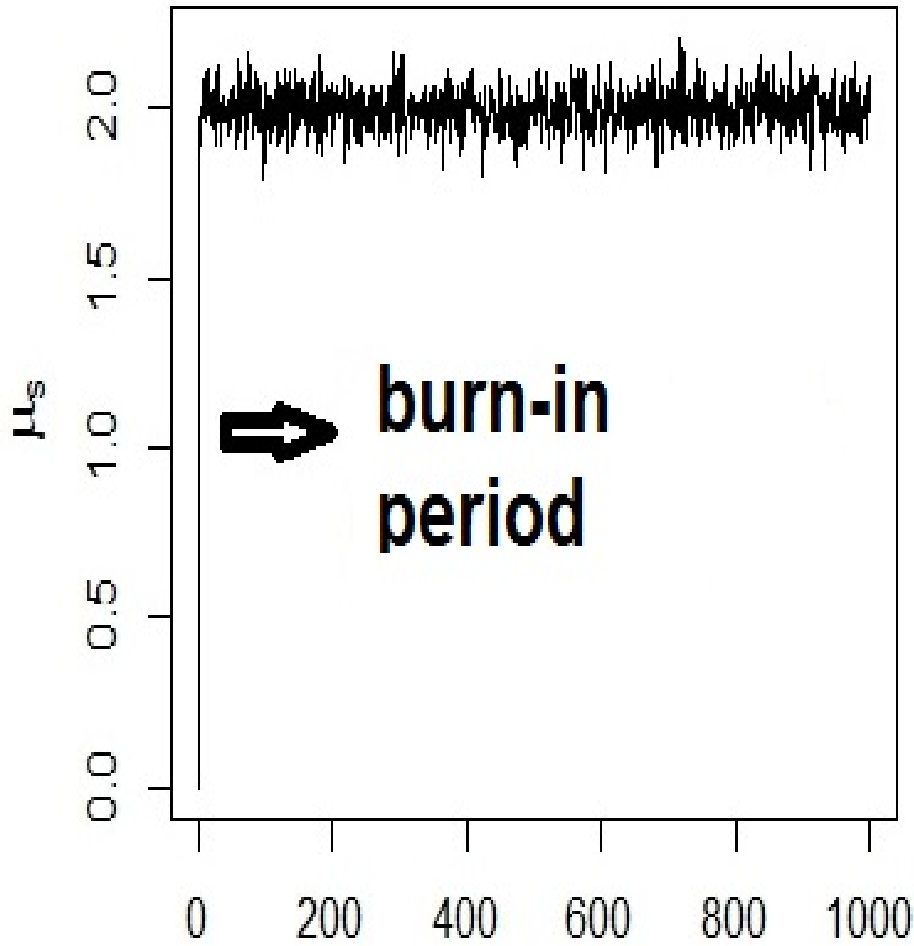
- Formal approaches to assessing convergence depend on whether
  1. We run a single MCMC - Gibbs sampler, MH algorithm - chain for a very large number of iterations S
  2. We run multiple - e.g., 3 - shorter MCMC chains

- In our previous examples we run a single chain
  - we started the chain from a single initial value for each parameter $\theta \in \Theta$
  - drew S samples from Θ, generating a sequence $\Theta^0, \Theta^1, \ldots, \Theta^S$

- But 2 is arguably more common in practice
  - e.g., running 3 chains, each one starting from different initial values
  - running each chain for S iterations
  - pooling together the draws from each chain - upon convergence - to compute means, variances, etc.

- In all cases, the usual practice is to:
  - use the first few iterations of the single/multiple chains as "burn-in"
  - assess convergence using the sample draws from the chains after the "burn-in" period

- In all cases, the usual practice is to:
  - use the first few iterations of the single/multiple chains as "burn-in"
  - assess convergence using the sample draws from the chains after the "burn-in" period



More formal convergence criteria - single chain

- **Geweke's criterion**, **Heidel's criterion**
  - compare the sample draws from different parts of the chain, after the burn-in period
  - check that the values do not differ dramatically

More formal convergence criteria - single chain

- **Geweke's criterion**, **Heidel's criterion**
  - compare the sample draws from different parts of the chain, after the burn-in period
  - check that the values do not differ dramatically
- For instance, using **Geweke's criterion** we would compare:
  - the initial 10% of the samples (after burn-in)
  - against the last 50% of the sampled values
  - using a t-type of test
  - "t-tests" outside the [-1.96, 1.96] range indicate lack of convergence

- Both the **Geweke** and **Heidel** criteria are readily available in R
  - "coda" package
  - commands: *geweke.diag*, *heidel.diag*
- Application: "Heidel criterion for Gibbs sampling, Normal distribution.R"
- Exercises:
  1. Check convergence using Geweke's criterion
  2. Use both the Heidel and Geweke criteria to check converge of the MH algorithm we used for ρ

More formal convergence criteria - multiple chains

- Gelman & Rubin's $\hat{R}$: compares the variability of sampled values
  - within each chain
  - and between chains
- More specifically:

1. Computes the average samples of θ in each chain c, $c = 1, 2, \ldots, C$ (e.g., $C = 3$)

$$\bar{\theta_c} = \frac{\sum_{s=1}^{S} \theta_c^s}{S}$$

2. Computes the average sampled values across chains:

$$\bar{\theta} = \frac{\sum_{c=1}^{C} \bar{\theta_c}}{C}$$

3. And from (1) and (2):

$$W = \frac{\sum_{c=1}^{C} \frac{\sum_{s=1}^{S} (\theta_c^S - \bar{\theta_c})^2}{S}}{C}$$

$$B = \frac{\sum_{c=1}^{C} (\bar{\theta_c} - \bar{\theta})^2}{S/(C-1)}$$

4. The convergence measure, $\hat{R}$, is given by:

$$\hat{R} = \frac{((S-1)/S)W + B/S}{W}$$

5. Convergence: $\hat{R} < 1.2$

- "coda" package
- command: *gelman.diag* '
- Application: "Gelman-Rubin diagnostic for Gibbs sampling, Normal distribution.R"

Assessing model fit in a Bayesian setting

- How to assess whether the model fits the data well?
- And how to compare two different models?
- No R^2 or pseudo-R^2: these are "frequentist" concepts
- Instead, Bayesians (e.g., Congdon (2009), Gelman (2007)) use **posterior predictive comparisons**

$$p(y_{replicated}|y_{observed}) = \int p(y_{replicated}|y_{observed}, \theta)p(\theta|y_{observed})d\theta$$

Assessing model "fit" in a Bayesian setting (cont.)

- Posterior predictive comparisons:
  1. simulate data from the estimated model parameters
  2. compare against the observed data
  3. use an overall fit measure to assess model fit
- Possible criteria to assess the posterior predictive comparisons:
  - % of correct predictions
  - whether the true data is in the 95% CI of the replicates
  - deviance
  - kurtosis, skewness (for normal data)

Example: Computing the % of correct predictions

- Application: "Posterior Predictions Probit.R"
- This script:
  1. Fits a simple (non-hierarchical) probit model
  2. Checks covergence
  3. Reports posterior summaries (means, 95% highest posterior density (HPD) intervals
  4. And computes the % of correct predictions
- Exercise: Compute other measures of goodness of fit based on the same model

Assessing model "fit" in a Bayesian setting (cont.)

- More formally, for each simulated value of the parameter s, generate a replicated data set $y^s_{replicated}$
- Choose a statistic D, and compare $D(y^s_{replicated})$ against $D(y_{observed})$
- Quantify the discrepancy
  - for instance, compute that % of correct predictions, or the proportion of times that the replicated y is above/below the "true" y
  - compute a "Bayesian p-value":

$$p = Pr\big(D(y_{replicated}) > D(y_{observed})\big)$$

- Systematic differences between replicate & actual data indicate model limitations

Comparing different models

- Two main tools:

  1. *DIC*: Deviance Information Criterion (most used)

     - Information criterion (like AIC or BIC), but specifically designed for MCMC simulations
     - In a nutshell: compares the expected log likelihood of the model against the likelihood at the posterior parameter means
     - **Always select the model with the lowest DIC**
     - "Rule of thumb": DIC differences larger than 3 provide overwhelming evidence in favor of the model with the lower value (Ntzoufras 2011)

Comparing different models (cont.)

2. *BF*: Bayes factors (less used, but comes "with" Stata)
   - Ratio of the likelihood of two models
   - Higher BF means more likely that the model is supported by the data
   - BF $>$ 10 provides strong evidence for the model with higher value (Kass & Raftery 1995)

Speeding up Bayesian Computations

- Bayesian models may be quite slow to run
- This is probably their main disadvantage
- Various approaches to deal with this issue
  - e.g., variational Bayesian inference, Hamiltonian Monte carlo
- But even MCMC algorithms can be accelerated
- We will mention 2 related approaches here
  - Rcpp: integrating R with C++
  - Rcpp + HPC

Integrating R with C++: Rcpp

- A first way to speed up MCMC algorithms is using Rcpp
- Rcpp allows "running C++ code" from R
  - learning to code in C++ from scratch is difficult
  - using Rcpp is much easier!
- Key reference: Eddelbuettel, Dirk. 2013. Seamless R and C++ Integration with Rcpp. New York, NY: Springer.

Comparison - Logit model in R and Rcpp

- Look at the R file: "comparison_logit.R"
- It compares the execution time of a Metropolis-Hastings algorithm fitted using:
  1. R
  2. Rcpp
- On average, the "pure" R code takes almost 6 times as much as the Rcpp code

Running Rcpp in the cluster

- To further accelerate execution, we can use an HPC cluster
- Exeter has the ISCA cluster avalailable:
  - 128 GB nodes
  - https://emps.exeter.ac.uk/computer-science/facilities/
- Key advantage for Bayesians: parallelization

- MCMC problems are "embarrasingly parallel"
    - If we have multiple cores, we can "send each MCMC chain" to a different core
- We don't need an HPC cluster for parallelization
    - multiple R packages do parallelization across clusters of the same computer
    - e.g., snow, doParallel
- However, typical desktop/laptop computer has 8 cores
    - with 3 chains per job, this means we can efficiently run at most 2 jobs in parallel
- With an HPC cluster, we can run tens/hundreds of multi-chain MCMC algorithms in parallel

- And the cost of paralellization is minimal
- Compare
    - "LogitRcpp.cpp"
    - and "LogitRcppHPC.cpp"
- Cost: 5 more lines of code
- Benefit: Cut execution time in less than half
- And this is a very simple problem (1 job, 3 chains)
    - Potential gains are huge!

Comparison of execution times

- Comparison: 100 replications ("estimations") of each model
- Each model runs 3 MCMC chains of length 1,000
- See "comparison_logit_parallel.R"

Table: Execution Times (in nanoseconds)

Model	Time	Ratio
R	3,657.47	13.17
Rcpp	589.61	2.12
Rcpp in ISCA	277.60	1.00

Additional readings

1. Eddelbuettel (2013). Seamless R andC++ Integration with Rcpp. New York, NY: Springer.
2. Gelman and Hill (2007): Data Analysis using Regression and Multilevel/Hierarchical Models
3. Gill (2008): Bayesian Methods: A Social and Behavioral Approach
4. Hahn, Eugene (2014): Bayesian Methods for Management and Business - Pragmatic Solutions for Real Problems.
5. Jackman (2009): Bayesian Analysis for the Social Sciences