STAT 7650 - Computational Statistics Lecture Slides

Markov Chain Monte Carlo

Elvan Ceyhan

Updated: April, 2025

AU

- Based on parts of: Chapters 7-8 in Givens & Hoeting (Computational Statistics), and Chapters 25-27 of Lange (Numerical Analysis for Statisticians)).

Outline

Motivation

- We know how to sample independent random variables from the target distribution $f(x)$, at least approximately.
- Monte Carlo uses these simulated random variables to approximate integrals.
- But the random variables don't need to be independent in order to accurately approximate integrals!
- MCMC constructs a dependent sequence of random variables that can be used to approximate the integrals just like for ordinary Monte Carlo.
- The advantage of introducing this dependence is that various general algorithms (and corresponding theory) are available to perform the required simulations (via MCMC).

Initial Remarks

- MCMC methods are powerful tools for sampling from complex probability distributions and have broad applicability.
- But its effectiveness depends on proper implementation and understanding of the problem context.

A Word of Caution

- It's crucial not to use MCMC — without a clear understanding of its applicability and limitations to your specific problem.
- That is, blindly applying MCMC (or any statistical tool) can lead to misleading results.

Understanding MCMC

- We will discuss some basics of Markov chains and MCMC.
- While MCMC is a mature field with many successful applications, ongoing research continues to explore its theoretical boundaries and practical implications.

Initial Remarks

Ongoing Research

- MCMC is an active area of research.
- Recent advances have significantly improved our understanding of MCMC convergence properties and efficiency.
- Innovations include adaptive MCMC, Hamiltonian Monte Carlo, and Variational Inference as a complementary approach.

Challenges and Frontiers in Research

- Despite advancements, challenges remain in ensuring efficiency and convergence in high-dimensional spaces.
- Research continues to focus on developing more robust, scalable, and efficient MCMC algorithms.
- Understanding the theoretical underpinnings of MCMC's convergence behavior in complex scenarios remains an open research area.

Outline

Markov Chains

A **Markov chain** is a sequence of random variables $\{X_1, X_2, \ldots\}$ with a specific type of dependence structure, where:

- The future state X_{n+1} given the past and present states $(X_1, \ldots, X_n)$ depends only on the present state X_n:

$$P(X_{n+1} \in B | X_1, \ldots, X_{n-1}, X_n) = P(X_{n+1} \in B | X_n)$$

- This property (called Markov Property) implies that the **probabilistic properties** of the chain are completely determined by:
  1. The initial distribution of X_0.
  2. The *transition distribution*, i.e., the distribution of X_{n+1} given X_n (usually the chain is assumed to be *homogeneous*, the transition distribution does not depend on n; i.e., the transition probabilities do not change over time).
- A sequence of independent rv's is a trivial special case of Markov chains.

Example: Simple Random Walk

- Let $U_1, U_2, \ldots$ be $\overset{iid}{\sim}$ (Discrete) Unif$(\{-1, 1\})$.
- Set $X_0 = 0$ and define $X_n = \sum_{i=1}^{n} U_i = X_{n-1} + U_n$.
- **Initial Distribution:** $P(X_0 = 0) = 1$.
- **Transition Distribution:** Given by

$$X_n = \begin{cases} X_{n-1} - 1 & \text{with probability } \frac{1}{2}, \\ X_{n-1} + 1 & \text{with probability } \frac{1}{2}. \end{cases}$$

- Despite its simplicity, the random walk is a foundational example in probability, with connections to more advanced concepts like Brownian motion.

Keywords[1] for Markov Chains

- **Recurrent State:** A state A is *recurrent* if a chain starting in A will eventually return to A with probability 1. A state is *null* if the expected time to return is infinite and *nonnull* if it is finite. A chain is recurrent if each state is recurrent.
- **Irreducible Markov Chain:** A Markov chain is *irreducible* if there is a positive probability that a chain starting in any state A can reach any other state B.
- **Aperiodic Markov Chain:** A Markov chain is *aperiodic* if, for any starting state A, there is no fixed number of steps in which the chain must return to A.
- **Ergodic Markov Chain:** An irreducible, aperiodic Markov chain with all states being nonnull recurrent is called *ergodic*.

[1] Not mathematically precise but serve for a foundational understanding

Limit Theory[2] in Markov Chains

- **Stationary Distribution:** A distribution f is stationary if $X_0 \sim f$ implies $X_n \sim f$ for all n.
- An ergodic Markov chain has at most one stationary distribution.
- **Ergodic Theorem:** For an ergodic Markov chain with stationary distribution f:
  - The limiting probability (distribution) of being in state B given starting in state A is given by:

$$\lim_{n\to\infty} P(X_{m+n} \in B | X_m \in A) = \int_B f(x)\,dx,$$

    for all states A, B, and times m.

[2]Simplified for clarity and are not mathematically precise

Limit Theory[3] in Markov Chains (Cont'd)

- **Ergodic Theorem (Cont'd):**
  - If $\varphi(x)$ is integrable, then the time average of $\varphi(X_t)$ converges almost surely to the expected value under f:

$$\frac{1}{n}\sum_{t=1}^{n}\varphi(X_t) \to \int \varphi(x)f(x)\,dx, \quad \text{as } n \to \infty$$

    with probability 1.

  This is a version of the famous *ergodic theorem*.
- **Central Limit Theorems for Markov Chains:** While not detailed here, there are central limit theorems applicable to Markov chains, extending some classic probability results to the Markov chain context.

[3] Simplified for clarity and are not mathematically precise

Outline

Why MCMC?

- In Monte Carlo simulations, our goal is to generate random variables with a specific distribution f.
- This task could be difficult or impossible to achieve exactly.
- **Solution:** MCMC is designed to construct an ergodic Markov chain for which f is its stationary distribution.
- Asymptotically, this chain will produce samples from f (but in practice, your samples resemble draws from f).
- By **Ergodic Theorem**, expectations with respect to f can be approximated by averages of samples from the Markov chain.
- Surprisingly, constructing and simulating a suitable Markov chain is quite manageable, which partly explains the popularity of MCMC methods.
- However, implementing MCMC methods comes with its set of practical and theoretical challenges...

Outline

Implementing the Metropolis-Hastings Algorithm

- Let $f(x)$ denote the target distribution pdf and
- $q(x|y)$ a conditional pdf for X, given $Y = y$; this pdf should be easy to sample from.
- Given X_0, the Metropolis-Hastings (MH) Algorithm produces a sequence of random variables by:
  1. Sample $X_t^* \sim q(x|X_{t-1})$.
  2. Compute

$$R = \min\left\{1, \frac{f(X_t^*)q(X_{t-1}|X_t^*)}{f(X_{t-1})q(X_t^*|X_{t-1})}\right\}.$$

  3. Set $X_t = X_t^*$ with probability R; otherwise, $X_t = X_{t-1}$.
- General `R` code for implementing the Metropolis-Hastings Algorithm is available on Canvas.

Choosing the Proposal Distribution

- The choice of proposal distribution $q(x|y)$ is crucial for the algorithm's performance but not easy.
- **Two General Strategies:**
  1. Use an *independent proposal* $q(x|y) = q(x)$, making X_t^* independent of X_{t-1} at each stage of the MH algorithm.
  2. Use a symmetric distribution $q(x|y) = q_0(||x - y||)$, which amounts to a *random walk proposal*.
- This aspect requires careful consideration for optimal performance.
- In the examples, I will just pick a proposal that seems to work reasonably well...

Convergence and Approximation in MCMC[4]

- Assuming the proposal distribution is adequately chosen, several properties about the sequence $\{X_t : t = 1, 2, \ldots\}$ can be established:
  - The chain is **ergodic**.
  - The target distribution f is the **stationary distribution**.
- As a result, the sequence converges to the stationary distribution. This means that for any integrable function $\phi(x)$, we can approximate integrals with sample averages.
- By running the simulation *long enough*, we can obtain arbitrarily good approximations.
- This presents an interesting opportunity for statisticians/data scientists: The ability to control the sample size for better approximation.

[4] Again, simplified and not mathematically precise

Example: Cosine Model

- **Problem Context:** From previous slides, recall the likelihood function defined as:

$$L(\theta) \propto \prod_{i=1}^{n} (1 - \cos(X_i - \theta)), \quad -\pi < \theta \leq \pi.$$

- **Observed Data:** Given as $(X_1, \ldots, X_n)$ in the provided code on Canvas.
- **Prior Distribution:** Assume θ follows a uniform distribution, $\text{Unif}(-\pi, \pi)$.
- Use **Metropolis-Hastings** to sample from the posterior:
  1. **Proposal Distribution:** $q(\theta'|\theta) = \text{Unif}(\theta'|\theta \pm 0.5)$.
  2. **Burn-in Period:** Set $B = 5000$ to allow the chain to stabilize.
  3. **Sample Size:** $M = 10000$

Example: Cosine Model - Visualization of Results:

- **Histogram with Posterior Density:** The left figure shows the histogram of the MCMC sample with the posterior density overlaid.
- **Trace Plot:** The right figure displays a trace plot of the chain, illustrating the convergence and mixing behavior of the Markov chain over iterations.

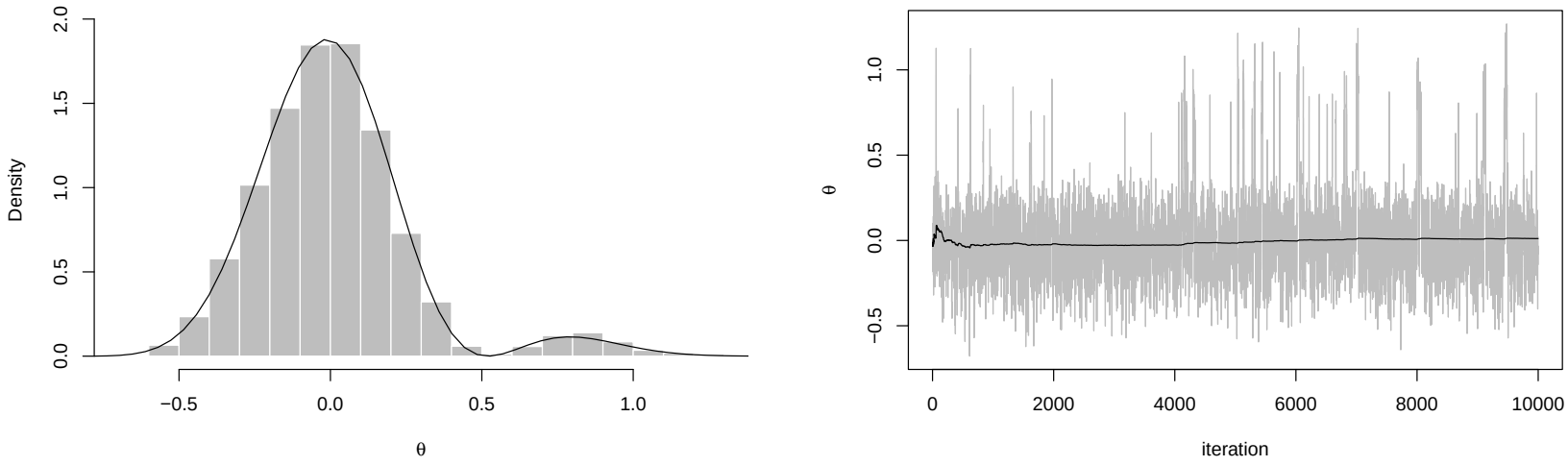

Figure 1: Left: Histogram of MCMC Sample. Right: Trace Plot of the Chain.

Example: Weibull Model

- **Data and Likelihood:** Data $X_1, \ldots, X_n$ iid from Weibull(λ, κ) distribution has likelihood:

$$L(\lambda, \kappa) = \lambda^{-\kappa n} \kappa^n \exp\left((\kappa - 1) \sum_{i=1}^{n} \log x_i - \lambda^{-\kappa} \sum_{i=1}^{n} x_i^{\kappa} \right).$$

- **Prior Density:** $\pi(\lambda, \kappa) \propto e^{-(\lambda + c\kappa)} \kappa^{b-1}$, for some constants b, c.
- **Posterior Density:** Proportional to

$$\lambda^{-\kappa n} \kappa^{n+b-1} \exp\left(\kappa \left(\sum_{i=1}^{n} \log x_i - c \right) - \lambda^{-\kappa} \sum_{i=1}^{n} x_i^{\kappa} - \lambda \right).$$

- **Goal:**
  Perform an informal Bayesian test of $H_0 : \kappa = 1$, where $\kappa = 1$ corresponds to the exponential distribution as a special case of the Weibull distribution.

Weibull Model Overview and Prior

Data and Likelihood Function: Given data $X_1, \ldots, X_n$ is iid from a Weibull(λ, κ) distribution, the pdf of X_i is given as

$$f(x_i|\lambda,\kappa) = \frac{\kappa}{\lambda}\left(\frac{x_i}{\lambda}\right)^{\kappa-1} e^{-(x_i/\lambda)^\kappa}, \quad x_i \geq 0,\ \lambda > 0,\ \kappa > 0.$$

and the likelihood function is given by:

$$L(\lambda,\kappa|x_1,\ldots,x_n) = \prod_{i=1}^{n} \frac{\kappa}{\lambda}\left(\frac{x_i}{\lambda}\right)^{\kappa-1} e^{-(x_i/\lambda)^\kappa}$$

$$= \lambda^{-\kappa n}\kappa^n \prod_{i=1}^{n} x_i^{\kappa-1} e^{-\left(\frac{x_i}{\lambda}\right)^\kappa} = \lambda^{-\kappa n}\kappa^n \prod_{i=1}^{n} \exp\left((\kappa-1)\log x_i - \lambda^{-\kappa} x_i^\kappa\right)$$

$$\propto \lambda^{-\kappa n}\kappa^n \exp\left(\kappa \sum_{i=1}^{n} \log x_i - \lambda^{-\kappa} \sum_{i=1}^{n} x_i^\kappa\right).$$

Prior and Posterior Density

Prior Density: The prior density for parameters λ and κ is assumed to be:

$$\pi(\lambda, \kappa) \propto e^{-(\lambda + c\kappa)} \kappa^{b-1},$$

where b and c are constants. This reflects our prior beliefs about the parameters before observing the data.

Posterior Density is proportional to:

$$\lambda^{-\kappa n} \kappa^{n+b-1} \exp\left(\kappa \left(\sum_{i=1}^{n} \log x_i - c \right) - \lambda^{-\kappa} \sum_{i=1}^{n} x_i^{\kappa} - \lambda \right).$$

Bayesian Testing

Goal: Bayesian Test for $\kappa = 1$
We aim to perform an informal Bayesian test of the null hypothesis $H_0 : \kappa = 1$. Testing $\kappa = 1$ is of particular interest as it corresponds to the Weibull distribution simplifying to an exponential distribution, which has a constant failure rate. This test allows us to evaluate if the exponential model is a suitable simplification given the data.

Interpretation
A Bayesian approach to hypothesis testing involves computing the posterior probability of the hypothesis given the data, which can be more informative than traditional *p*-values. It provides a direct probability statement about the hypothesis.

Example: Weibull Model (Cont'd)

- **Data Source:** Problem 7.11 in Ghosh et al (2006).
- **Sampling Method:** Use MH to sample from the posterior of (λ, κ).
- **Proposal Distribution:** $(\lambda', \kappa')|(\lambda, \kappa) \sim \text{Exp}(\lambda) \times \text{Exp}(\kappa)$.
- **Prior Parameters:** $b = 2$ and $c = 1$; Burn-in $(B) = 1000$ and Sample Size $(M) = 10000$.
- **Visualization:** Histogram of the marginal posterior of κ.
- **Evaluation:** Is an exponential model $(\kappa = 1)$ reasonable based on the marginal posterior?

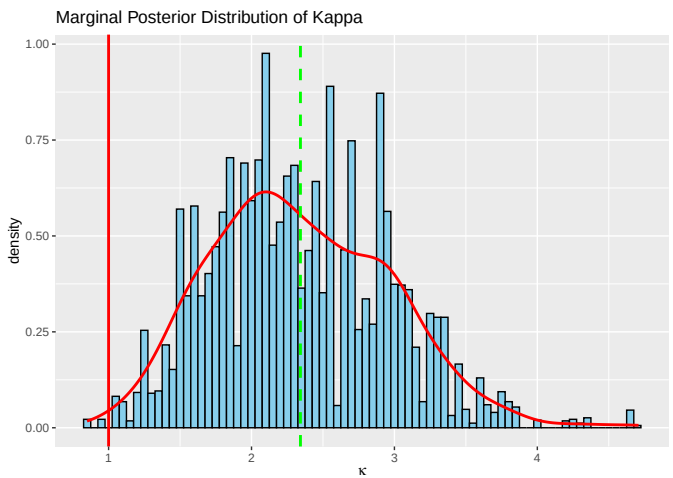

Figure 2: Marginal Posterior of κ

Example: Logistic Regression

- **Background:** Based on Examples 1.13 and 7.11 in Robert & Casella's book.
- **Incident:** In 1986, the Challenger space shuttle exploded due to an "o-ring" failure, possibly influenced by the cold temperature (31°F).
- **Goal:** Analyze the relationship between temperature and o-ring failure.
- **Approach:** Fit a logistic regression model to understand the impact of temperature on the probability of o-ring failure.

Example: Logistic Regression (Cont'd)

- **Model Specification:**
  - Model: $Y|x \sim \text{Ber}(p(x))$, where x represents temperature.
  - Failure probability, $p(x)$, modeled as:

$$p(x) = \frac{\exp(\beta + \gamma x)}{1 + \exp(\beta + \gamma x)}.$$

- **Model Fitting:**
  - Fitted using `glm` in `R` with available data.
- `Coefficients:`

	Estimate	Std. Error	z value	Pr(>|z|)
Intercept	15.0429	7.3786	2.039	0.0415 *
x	-0.2322	0.1082	-2.145	0.0320 *

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 .  0.1 1
```

- **Observation:** Probability of failure at 31°F is approximately 0.999!!!

Example: Logistic Regression (Cont'd)

- **Bayesian Analysis:**
  - Can also do a Bayesian analysis of this logistic model.
  - Use MH to obtain samples from the posterior of (β, γ).
  - These samples can be used to approximate the posterior distribution of $p(x_0)$ for any fixed x_0, e.g., 65°F and 31°F.
- **Details:**
  - Prior and proposal construction details are available in the `R` code posted on Canvas.
- **Posterior Distributions:**
  - Plots for $p(65)$ and $p(31)$ (see next slide) demonstrate the density of the posterior probabilities at these temperatures.

Posterior Plots for $p(65)$ and $p(31)$

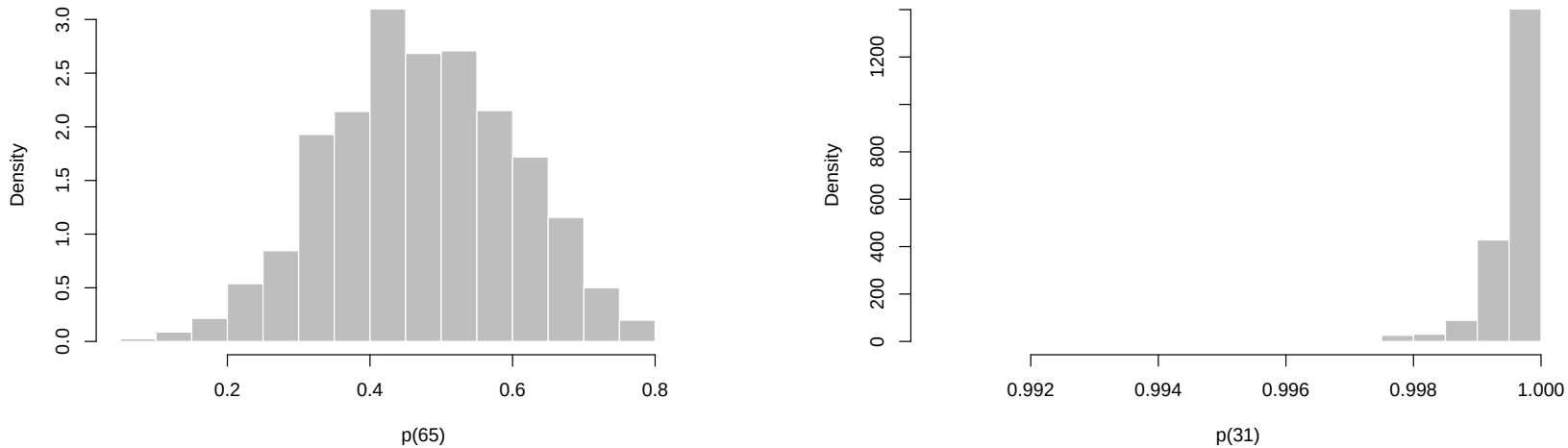

Figure 3: Posterior (predictive) distributions of $p(65)$ and $p(31)$.

Outline

Setup for Multivariate Distributions

- **Problem Context:** Given a multivariate target distribution f.
- **Challenge with MH:** Applying MH to multivariate distributions introduces challenges, particularly in constructing effective proposals across multiple dimensions.
- **Proposed Solution:** Sample one dimension at a time to mitigate the complexity of multi-dimensional proposal construction.
- **Key Question:** How can we ensure that such sampling accurately approximates the target distribution, especially in the limit?
- **Gibbs Sampler:** Identified as an optimal approach for this task, the Gibbs sampler systematically samples each dimension, effectively approximating the multivariate target distribution.

Details on the Gibbs Sampler

- **Target Distribution:** Consider a trivariate target distribution $f(\mathbf{x}) = f(x_1, x_2, x_3)$.
- **Full Conditionals:** Assume that we can express the distribution in terms of its full conditionals:

$$f(x_1|x_2, x_3), \quad f(x_2|x_1, x_3), \quad f(x_3|x_1, x_2)$$

- **Sampling Process:** Assuming we can sample from these conditionals, the Gibbs sampler iterates as follows:

$$X_1^{(t)} \sim f(x_1|X_2^{(t-1)}, X_3^{(t-1)})$$
$$X_2^{(t)} \sim f(x_2|X_1^{(t)}, X_3^{(t-1)})$$
$$X_3^{(t)} \sim f(x_3|X_1^{(t)}, X_2^{(t)})$$

- Each step involves sampling from the conditional distribution of one variable, holding the others at their current values.

Details on Gibbs Sampler (Cont'd)

- **Markov Chain:** The sequence generated by the Gibbs sampler forms a Markov chain.
- **Relationship to MH:**
  - The Gibbs sampler is a special case of MH!
  - It can be viewed as an MH sequence that updates one component of $\mathbf{X}$ at a time.
- **Acceptance Probability:**
  - The acceptance probability for updates in the Gibbs sampler is exactly 1.
  - This characteristic explains the absence of an accept/reject step in the Gibbs sampler.
- **Convergence:**
  - Since the Gibbs sampler is a special kind of MH, the convergence properties of MH also apply to Gibbs.

Example: Bivariate Normal with Gibbs Sampling

- A simple Gibbs sampling example: Sampling from a bivariate normal distribution.
- Suppose $\mathbf{X} = (X_1, X_2)$ is bivariate normal with $\mu_1 = \mu_2 = 0$, $\sigma_1 = \sigma_2 = 1$, and correlation ρ.
- **Full conditionals:** The full conditionals are straightforward to derive in this context.
- Gibbs Sampling Steps:

$$X_1^{(t)} \sim N(\rho X_2^{(t-1)}, 1 - \rho^2)$$
$$X_2^{(t)} \sim N(\rho X_1^{(t)}, 1 - \rho^2)$$

- While not as efficient as direct sampling from the bivariate normal distribution, Gibbs sampling performs well in this scenario.

Example: Many-Normal-Means Model (Hierarchical Bayes)

- **Model Specification:**
  - Consider $X_i \overset{ind}{\sim} N(\theta_i, 1)$, for $i = 1, \ldots, n$.
- **Hierarchical Prior Distribution:**
  - $\theta_1, \ldots, \theta_n | \sigma^2 \overset{iid}{\sim} N(0, \sigma^2)$; $\sigma^{-2} \sim \text{Gamma}(a, b)$.
- **Full (Posterior) Conditionals:** Takes some work[5], but it can be shown that the full conditionals are
  - $$\theta_i | (x_i, \sigma^2) \overset{ind}{\sim} N\left(\frac{x_i}{1+\sigma^2}, \frac{1}{1+\sigma^2}\right), \text{ for } i = 1, \ldots, n.$$
  - $$\sigma^{-2} | (\boldsymbol{\theta}, \mathbf{x}) \sim \text{Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2}\sum_{i=1}^{n} \theta_i^2\right).$$
- **Gibbs Sampler Implementation:** The Gibbs sampler can be straightforwardly implemented using these full conditionals.

[5] It can be shown based on standard conjugate priors.

Example: Many-Normal-Means Model (Cont'd)

- Suppose the goal is to estimate $\|\boldsymbol{\theta}\|^2 = \sum_{i=1}^{n} \theta_i^2$.
  - The MLE $\|\mathbf{X}\|^2$ performs poorly for this purpose.
  - The Bayes estimator, $\mathbf{E}(\|\boldsymbol{\theta}\|^2|\mathbf{X})$, provides a superior alternative and can be evaluated using the Gibbs sampler.
- **Rao-Blackwellization:**
  - Using the Rao-Blackwellized estimator for $\mathbf{E}(\theta_i^2|\mathbf{x})$ can further reduce variance.
- **Simulation Study:**
  - Objective: Compare the Bayes estimator with MLE.
  - Settings: $n = 10$, $\boldsymbol{\theta} = (1, 1, \ldots, 1)$, 1000 repetitions, 5000 Monte Carlo simulations, $a = b = 1$.
  - Results:

mle_mse	bayes_mse
180.1721	32.93027

Example: Many-Normal-Means Model (Details)

- **Model Specification:**
  - Consider a sequence of observations $X_1, X_2, \ldots, X_n$ where each X_i is normally distributed with mean θ_i and variance 1. Mathematically, this is expressed as $X_i \overset{ind}{\sim} N(\theta_i, 1)$ for $i = 1, \ldots, n$. This specification suggests that each observation has its own unique mean but shares the same error variance.
- **Hierarchical Prior Distribution:**
  - The means θ_i are assumed to be drawn from a common normal distribution, indicating a shared underlying process but allowing individual differences. Specifically, $\theta_1, \ldots, \theta_n | \sigma^2 \overset{iid}{\sim} N(0, \sigma^2)$. The variance parameter σ^2 is itself a random variable, following an Inverse Gamma distribution, $\sigma^{-2} \sim \text{Gamma}(a, b)$, providing a flexible prior that can adapt based on the observed data.

Example: Many-Normal-Means Model (Details)

- **Full Conditionals:**
  - The posterior distributions or full conditionals for each parameter are derived using Bayesian conjugacy. For θ_i, given the variance σ^2 and the data x_i, the posterior is also normal:

$$\theta_i|(x_i, \sigma^2) \overset{ind}{\sim} N\left(\frac{x_i}{1+\sigma^2}, \frac{1}{1+\sigma^2}\right), \text{ for } i = 1, \ldots, n.$$

  - The posterior for σ^{-2}, conditional on all θ_i and the data $\mathbf{x}$, is a Gamma distribution:

$$\sigma^{-2}|(\boldsymbol{\theta}, \mathbf{x}) \sim \text{Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2}\sum_{i=1}^{n}\theta_i^2\right).$$

- **Gibbs Sampler Implementation:**
  - The Gibbs sampler is particularly useful here due to the tractability of the full conditionals. Step-by-step, this involves alternately updating θ_i and σ^2 using their respective distributions, facilitating efficient Bayesian inference.

Example: Many-Normal-Means Model (Details)

Estimating $\|\boldsymbol{\theta}\|^2$:

- The objective is to estimate the squared norm of the parameter vector $\boldsymbol{\theta}$, $\|\boldsymbol{\theta}\|^2 = \sum_{i=1}^{n} \theta_i^2$. This quantity measures the overall magnitude of the effects in many statistical models.
- **Maximum Likelihood Estimator (MLE):** $\|\mathbf{X}\|^2$ is a natural estimator, but it tends to be biased upwards, especially in small samples or when variance is high.
- **Bayesian Estimator:** $\mathbf{E}(\|\boldsymbol{\theta}\|^2|\mathbf{X})$ leverages the posterior distributions of θ_i and provides a more reliable estimate by incorporating prior information and the uncertainty inherent in the observed data.

Rao-Blackwellization: Used to refine the estimator for $\mathbf{E}(\theta_i^2|\mathbf{x})$, thereby reducing the variance of the estimator compared to the unconditioned estimator. This method effectively integrates out the Monte Carlo noise associated with the Gibbs sampling, leading to a more stable estimate.

Example: Many-Normal-Means Model (Details)

- **Simulation Study:**
  - **Objective:** The goal of this study is to empirically compare the performance of the Bayes estimator against the MLE in terms of mean squared error (MSE).
  - **Settings:** The simulation setup includes $n = 10$ variables, all θ values set to 1 (representing a simplified scenario where all effects are equal), 1000 repetitions to ensure stability of the results, and 5000 Monte Carlo simulations to approximate the MSE accurately.
  - **Results:**

MLE MSE	**Bayes MSE**
180.1721	32.93027

  - These results demonstrate that the Bayesian estimator substantially outperforms the MLE in this setting, having a much lower MSE and thus providing more accurate and reliable estimates.

Example: Capture-Recapture Study

- **Context:** Example 7.6 in G&H. Study designed to estimate the population size (N) of fur seal pups in a coastal region in NZ, where N is unknown.
- **Capture-Recapture Study:**
  - Conducted over n occasions, with fur seal pups being caught, marked, and then returned to the ocean (in NZ).
  - At each occasion $i = 1, \ldots, n$:
    - C_i = number of pups caught at time i.
    - R_i = number of "recaptures" at time i.
    - $C_i - R_i$ = number of new pups caught at time i.
  - Define $U_i = \sum_{j=1}^{i} (C_j - R_j)$, the cumulative count of new pups caught up to time i.
- **Model Assumptions:**
  - The model assumes independent binomial sampling for the capture-recapture process.

Example: Capture-Recapture Study (Cont'd)

- **Binomial Success Probabilities:**
  - Introduce $\omega_1, \omega_2, \ldots, \omega_n$ as the binomial success probabilities.
- **Likelihood for $(N, \boldsymbol{\omega})$:**

$$L(N, \boldsymbol{\omega}) =$$

$$\prod_{i=1}^{n} \binom{U_{i-1}}{R_i} \omega_i^{R_i} (1-\omega_i)^{U_{i-1}-R_i} \binom{N-U_{i-1}}{C_i-R_i} \omega_i^{C_i-R_i} (1-\omega_i)^{N-U_{i-1}-C_i+R_i}$$

$$= \prod_{i=1}^{n} \binom{U_{i-1}}{R_i} \binom{N-U_{i-1}}{C_i-R_i} \omega_i^{C_i} (1-\omega_i)^{N-C_i}$$

$$= \frac{N!}{(N-U_n)!} \times \prod_{i=1}^{n} \binom{U_{i-1}}{R_i} \omega_i^{C_i} (1-\omega_i)^{N-C_i}.$$

- **Priors:**
  - $N \sim \text{Poi}(m)$ for the unknown population size.
  - $\omega_i \overset{iid}{\sim} \text{Beta}(a, b)$ for the success probabilities.

Example: Capture-Recapture Study (Cont'd)

- **Posterior Distribution of $(N, \boldsymbol{\omega})$:**

$$\propto \frac{N!}{(N-U_n)!}\frac{m^N}{N!} \times \prod_{i=1}^{n} \binom{U_i - 1}{R_i} \omega_i^{C_i+a-1}(1-\omega_i)^{N-C_i+b-1}.$$

- **Full Conditionals:** To run a Gibbs sampler, we need the full conditionals.
  - For $\omega_i|(N, \text{data})$ independently follows:

$$\omega_i|(N, \text{data}) \sim \text{Beta}(a + C_i, b + N - C_i), \quad i = 1, \ldots, n.$$

  - For $N|(\boldsymbol{\omega}, \text{data})$:

$$N|(\boldsymbol{\omega}, \text{data}) \sim U_n + \text{Poi}\left(m \times \prod_{i=1}^{n}(1-\omega_i)\right).$$

- **Gibbs Sampler Implementation:**
  - With these full conditionals, implementing the Gibbs sampler is straightforward.

Example: Probit Regression

- **Model Specification:** Observations $Y_i \overset{ind}{\sim} \text{Ber}(p_i = \Phi(\mathbf{x}_i^\top \boldsymbol{\beta}))$, Bernoulli distribution with probability p_i for $i = 1, \ldots, n$, where Φ denotes the standard normal cdf.
- **Prior for $\boldsymbol{\beta}$:** A normal prior.
- **Gibbs Sampling Challenge:** It is not immediately clear how to implement Gibbs sampling to obtain samples from the posterior distribution of $\boldsymbol{\beta}$.
- **Introduction of "Missing Data":**
  - Recall from EM slides that the model can be simplified by introducing some "missing data."
  - The conditional distribution of the missing data, given the observed data and $\boldsymbol{\beta}$, constitutes one part of the full conditionals.
  - The model for the complete data is, by construction, nice, simplifying the other part of the full conditionals.

Example: Probit Regression (Cont'd)

- **Missing Data:**

$$Z_i \overset{ind}{\sim} N(\mathbf{x}_i^T \boldsymbol{\beta}, 1) \text{ and } Y_i = I(Z_i > 0), \quad i = 1, \ldots, n.$$

- **Full Conditionals:**
  - Distribution of $\boldsymbol{\beta}$, given $(\mathbf{Y}, \mathbf{Z})$, depends only on $\mathbf{Z}$ and is straightforward due to the conjugate normal prior for $\boldsymbol{\beta}$.
  - Distribution of $\mathbf{Z}$, given $(\mathbf{Y}, \boldsymbol{\beta})$, is a truncated normal.
- **Gibbs Sampler Construction:**
  - While exact details are not provided here, constructing a Gibbs sampler for this setup is manageable[6].
  - See Section 8.3.2 in Ghosh et al (2006) for a detailed guide.

[6] The only potential difficulty is simulating from a truncated normal when the truncation point is extreme, but remember we have talked about extreme normal tail probabilities before...

Example: Probit Regression (Detailed Take)

Model Specification:

- **Probabilistic Framework:** Each observation Y_i is independently modeled as a Bernoulli distributed random variable, $Y_i \overset{\text{ind}}{\sim} \text{Ber}(p_i)$, where $p_i = \Phi(\mathbf{x}_i^\top \boldsymbol{\beta})$. Here, Φ represents the cumulative distribution function (cdf) of the standard normal distribution, mapping the linear predictor $\mathbf{x}_i^\top \boldsymbol{\beta}$ to a probability between 0 and 1.
- **Predictors and Parameters:** The model incorporates n observations, with each observation i having an associated vector of predictors $\mathbf{x}_i$ and a shared parameter vector $\boldsymbol{\beta}$.

Prior Distribution for $\boldsymbol{\beta}$:

- **Normal Prior:** The parameter vector $\boldsymbol{\beta}$ is assumed to follow a multivariate normal distribution. This prior reflects our beliefs about the parameter values before observing any data.

Example: Probit Regression (Detailed Take)

Challenges in Gibbs Sampling:

- **Complexity in Posterior Sampling:** Directly sampling from the posterior distribution of $\boldsymbol{\beta}$ using Gibbs sampling is challenging due to the nonlinear transformation involved in the Bernoulli probabilities through the normal cdf.

Introduction of Missing Data to Simplify the Model:

- **Latent Variable Approach:** By introducing latent variables, we can transform the probit regression into a model with a simpler complete-data likelihood, which is more amenable to the application of Gibbs sampling.
- **Conditional Distributions:** The missing data approach leverages the latent variable structure to separate the complex dependency into simpler, conditional distributions that are easier to sample from using standard Gibbs sampling techniques.

Example: Probit Regression (Continued)

Missing Data Formulation:

- **Latent Variable Definition:** Each latent variable Z_i for $i = 1, \ldots, n$ is independently drawn from a normal distribution centered around the linear combination of predictors, $Z_i \overset{\text{ind}}{\sim} N(\mathbf{x}_i^T \boldsymbol{\beta}, 1)$.
- **Observation Model:** The observed binary outcomes Y_i are determined by the sign of Z_i, specifically $Y_i = I(Z_i > 0)$, where I is an indicator function that converts the latent variable into a binary outcome.

Example: Probit Regression (Continued)

Full Conditional Distributions:

- **For β:** Given the latent variables $\mathbf{Z}$, the distribution of β conditional on $\mathbf{Z}$ and $\mathbf{Y}$ relies solely on $\mathbf{Z}$. It is a normal distribution, which simplifies due to the conjugate relationship between the normal prior and the normal likelihood.
- **For Z:** The distribution of $\mathbf{Z}$ given β and $\mathbf{Y}$ is a truncated normal, where the truncation limits are determined by the corresponding Y_i values (truncated below zero if $Y_i = 0$ and above zero if $Y_i = 1$).

Gibbs Sampler Implementation:

- **Implementation Strategy:** The Gibbs sampler alternates between sampling β from its full conditional using the normal distribution and $\mathbf{Z}$ from the truncated normal distribution.

Example: Dirichlet Process Mixture Model

- **Context:** In Bayesian nonparametrics, the Dirichlet process mixture (DPM) model probably the most widely used.
- **Model Flexibility:**
  - Offers a flexible approach for density estimation.
  - Utilizes a normal mixture density without specifying component means, variances, or the number of components.
- **Main Challenge:**
  - Traditional mixture models struggle with choosing the optimal number of components.
  - DPM models select the number of components automatically, addressing this challenge.
- **Computational Feasibility:** Despite its "nonparametric" label, computations for DPM models are manageable, typically involving a Gibbs sampler.

- **References:**
  - Escobar & West (JASA, 1995) present the simplest algorithm for DPM.
  - Kalli et al. (Stat Comp, 2011) propose an efficient slice sampler for DPM.

Application: Dirichlet Process Mixture Model

- **Application:** Implementing the slice sampler from Kalli et al to fit a normal mixture model to galaxy data.
- **Visualization:**
  - Density estimation and posterior mean comparison with kernel density estimates.
  - Analysis of the number of components and their probability distribution.

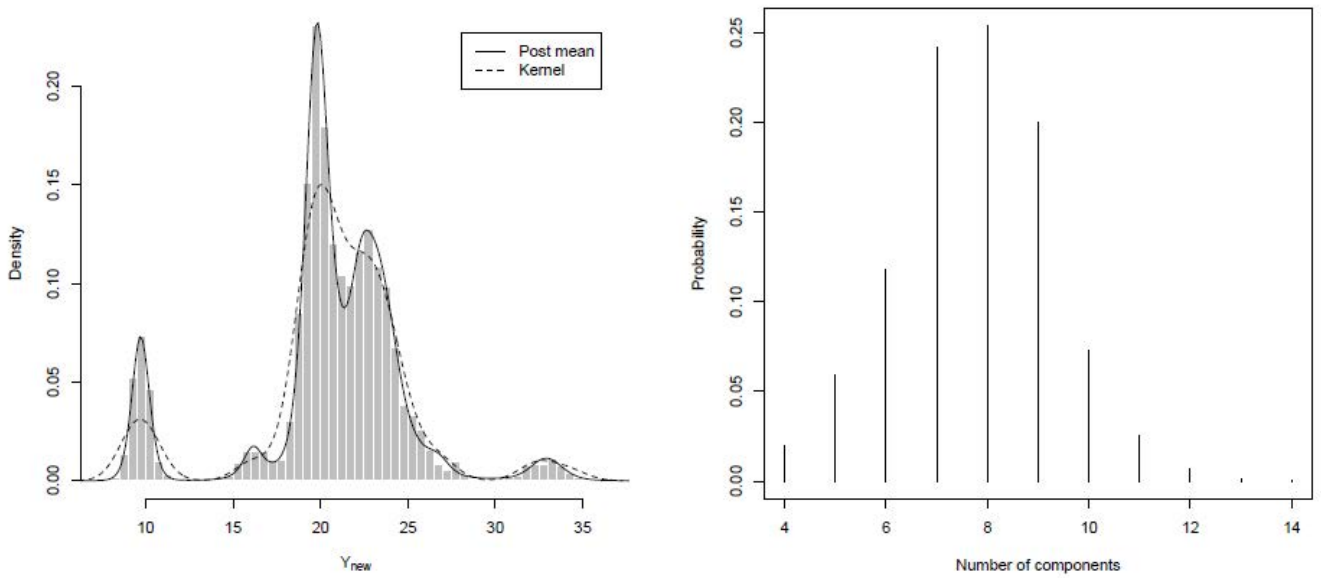

Figure 4: Left: Density Estimation with DPM. Right: Number of Components Probability.

Outline

Diagnostic Plots: Sample Path (Trace) Plot

- **Purpose:** To reveal any residual dependence after the burn-in period.
- **Idea:**
  - A sample path of iid samples should show no trend.
  - Minimal trend in our sample plot suggests we can treat samples as independent.
- **Usage:** Analyze the trace plot for absence of trends or patterns, indicating successful burn-in.

Diagnostic Plots: Autocorrelation (acf) Plot

- **Purpose:** To assess the dependence structure along the chain.
- **Method:** Plotting sample correlation of $\{(X_t, X_{t+r}) : t = 1, 2, \ldots\}$ as a function of the "lag" r.
- **Desired Outcome:** Rapid decay in the autocorrelation plot, indicating weak dependence along the chain.
- **Actions for Non-Convergence:** If the trace and acf plots suggest the chain has not converged to stationarity, consider running the chain longer or applying modifications such as transformations or "thinning".

Other Considerations in MCMC Convergence

- **Rate of Convergence:** Practical/theoretical convergence rates can vary with parametrization. (Refer to homework for examples.)
- **Community Consensus:** No unified agreement exists in the statistical community regarding the optimal number of chains, length of burn-in, etc.
- **Perspectives on Chain Management:**
  - Charles Geyer (University of Minnesota) advocates for running a single, long chain. (See his "rants" for more insight.)
  - Gelman & Rubin recommend running several shorter chains from different starting points, providing a diagnostic test in their textbook.

Outline

Remarks on MCMC Methods

- **Power of MCMC Methods:** They offer general procedures for solving a variety of important problems.
- **(Classical) Software Implementations:**
  - R's `mcmc` package for random walk Metropolis-Hastings.
  - SAS's PROC MCMC has similar capabilities.
  - BUGS (Bayesian inference Using Gibbs Sampling) for Gibbs sampling.
  - See next slide for state-of-the-art current implementations.
- **Caveat:** Blind reliance on software without understanding the underlying methodology and its appropriateness for your specific problem can lead to misleading results.
- **Convergence Diagnostics:** Essential to assess convergence through diagnostics before utilizing simulation results for inference.

Modern MCMC Software (for Bayesian Inference)

- **Stan**: Advanced statistical modeling platform supporting methods like Hamiltonian Monte Carlo (HMC) and NUTS (No-U-Turn Sampler, an extension of the HMC method) for efficient analysis of complex models.
- **PyMC**: Python library for probabilistic programming, with PyMC3 offering automatic differentiation and GPU acceleration for advanced algorithms.
- **Turing.jl**: Flexible and fast Julia library for Bayesian inference, supporting diverse sampling methods including HMC and NUTS.
- **JAGS**: Extensible Gibbs sampler for complex Bayesian analysis, offering a versatile alternative to BUGS.
- **R's `brms` package**: User-friendly and high-level interface primarily built on top of Stan, for fitting Bayesian regression models using R's formula syntax, simplifying Bayesian

Integrating MCMC Methods

- Our discussion primarily centered on the simpler forms of MCMC methods.
- **Integration of Methods:**
  - Methods like Metropolis-Hastings (MH) and Gibbs sampling are not mutually exclusive and can be combined for enhanced flexibility and efficiency.
  - For instance, an MH step can be incorporated within a Gibbs cycle sampling to address full conditionals that are challenging to sample from directly.
- **Further Reading:** The book by Robert & Casella provides insights into more advanced MCMC techniques, including various combinations of standard methods.