A simple introduction to Markov Chain Monte-Carlo

HSS552 20241105 Reading #5

van Ravenzwaaij, D., Cassey, P. & Brown, S.D. A simple introduction to Markov Chain Monte–Carlo sampling. Psychon Bull Rev 25, 143–154 (2018). https://doi.org/10.3758/s13423-016-1015-8

20244261 Jiyoung Baek jiyoung@kaist.ac.kr

INDEX

00

Before Start

Access R code

In this link :

https://github.com/BaekJiyoungg/HSS552

PPT materials

In this link :

https://github.com/BaekJiyoungg/HSS552

Lecture 7 Review

This concept has been already mentioned and studied.

00 Goals

What MCMC is? (Index 1)	What it can be used for (index 2)	Benefits & Limitations (Index 1,3)	Expansion to Multivariate Distribution (Index 4& 5 & 6)

01

Introduction

- **What MCMC is?**
- **What benefits?**
- **Principles**

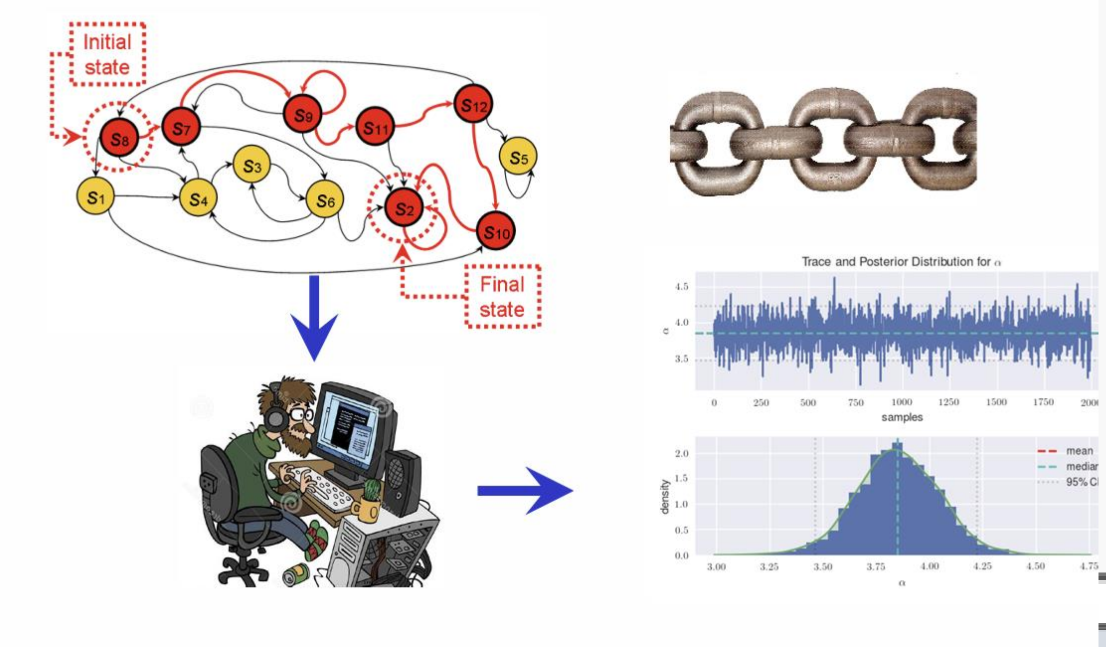
01 What MCMC is?

MCMC (Markov Chain Monte-Carlo)

→ One of Sampling methods

Markov Chain(Lec2)

Random samples
New sample is affected by only the previous one.

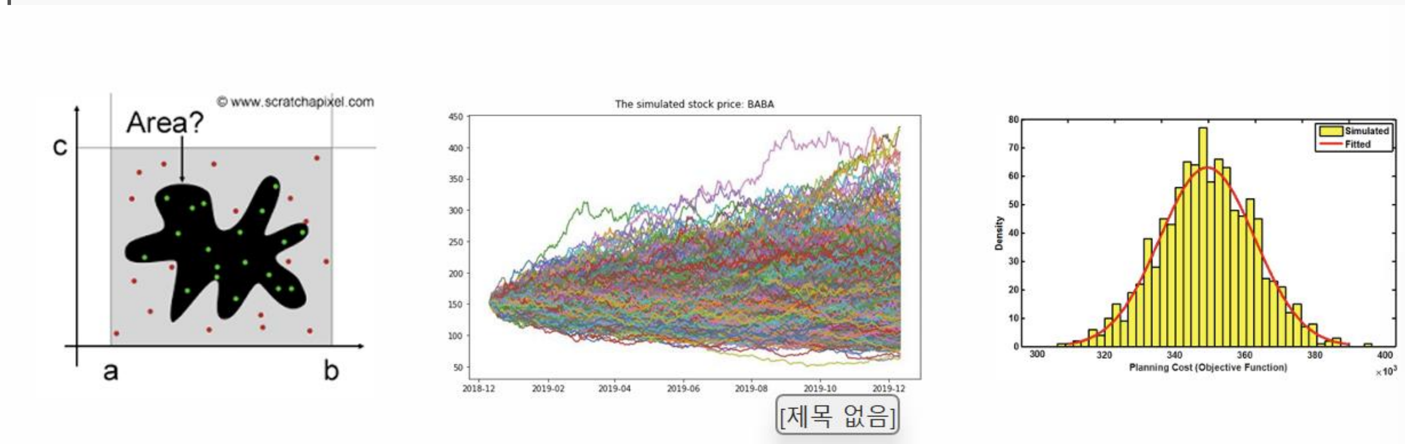


First paper on MCMC:
Metropolis, N, Rosenbluth, A W, Rosenbluth, M N, Teller, A H & Teller, E (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics, 21*, 1087-1092.
31

+

Monte-Carlo(Lec2)

Draw a large number of random samples from a normal distribution's equations



01

Why MCMC?

MCMC is useful in Bayesian inference.

- **Why? Focus on posterior distributions, which are often difficult to work with via analytic examination.**
    - ▶ **approximates aspects of posterior distributions**
- **Solution for when an analytical expression is inaccessible.**
    - **Drawing a sequence of samples from the posterior, and examining their mean range, and so on.**

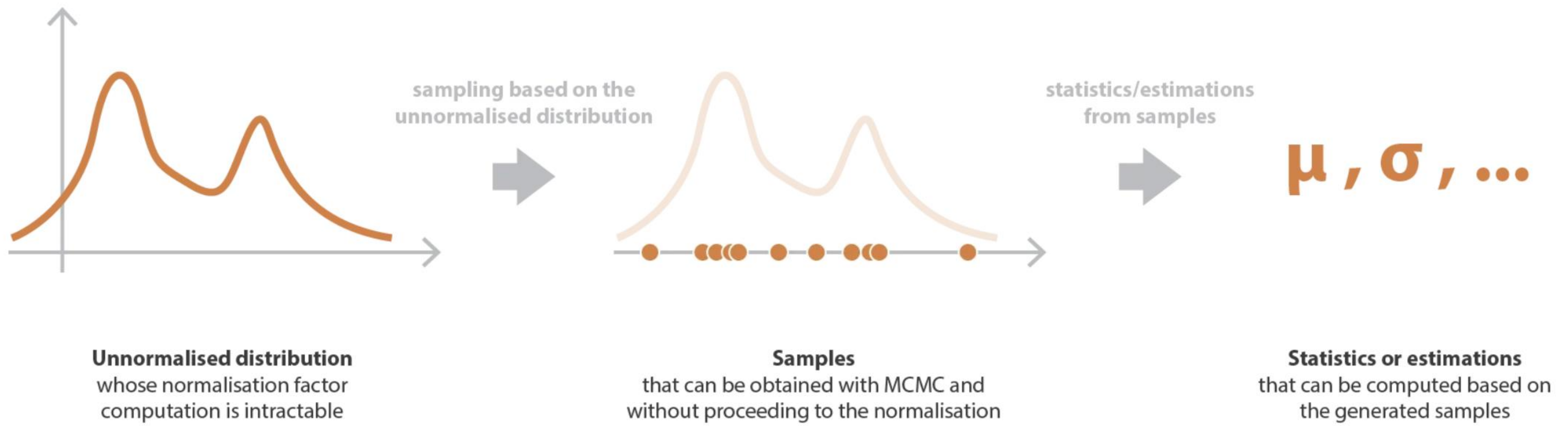


Illustration of the sampling approach (MCMC).

"Bayesian inference has benefited greatly from the power of MCMC."

02

Example

- In-class test
- Target distribution
- Metropolis Algorithm

02 In-class example

KEYWORD

01 Goal : the mean test score (μ)

Target distribution

: $X \sim N(\mu, \sigma^2 = 15^2)$

→ Univariate random variable

02 Metropolis algorithm

- One value that might be plausibly drawn from the distribution.
- Symmetric

03 Two steps

1. A proposal for the new sample by adding a small random perturbation to the most recent sample
2. This new proposal is accepted or rejected

04 Limitation

- Symmetric
- The initial guess is important

+)

Basic Statistic(SKIP)

KEYWORD

01

Univariate distribution

of variables = 1

02

Joint distribution

The joint probability mass function of two discrete random variables X and Y is defined as

P(x, y) = P(X=x. Y=Y)

03

Multivariate distribution

- Underlying random structure of vector of random variables, which is called random vector
- N(observation) x P(variables)

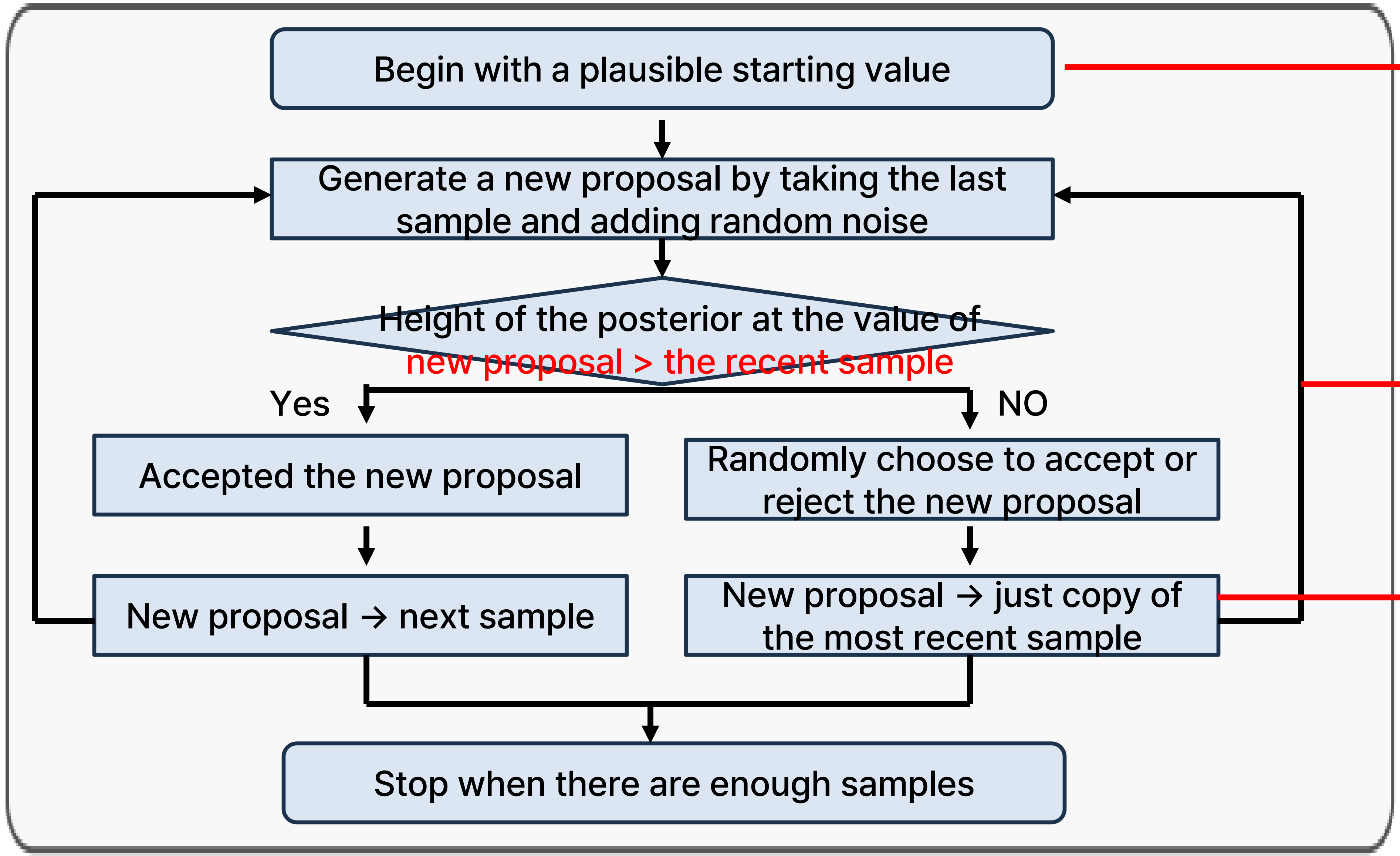
04

Symmetric distribution

Occurs when the values of variables appear at regular frequencies and often the mean, median, and mode all occur at the same point.

02 In-class example

< Metropolis Algorithm >



" Since the initial guess might be very wrong, the first part of the Markov chain should be ignored
: these early samples cannot be guaranteed to be drawn from the target distribution. "

" The likelihood values to accept or reject the new proposal must accurately reflect the density of the proposal in the target distribution. "

" The proposal distribution should be symmetric
If not symmetric → Metropolis-Hastings "

03

Limitation

- convergence
- Burn-in
- Tuning parameter

03 In-class example - limitation

< Important Findings >

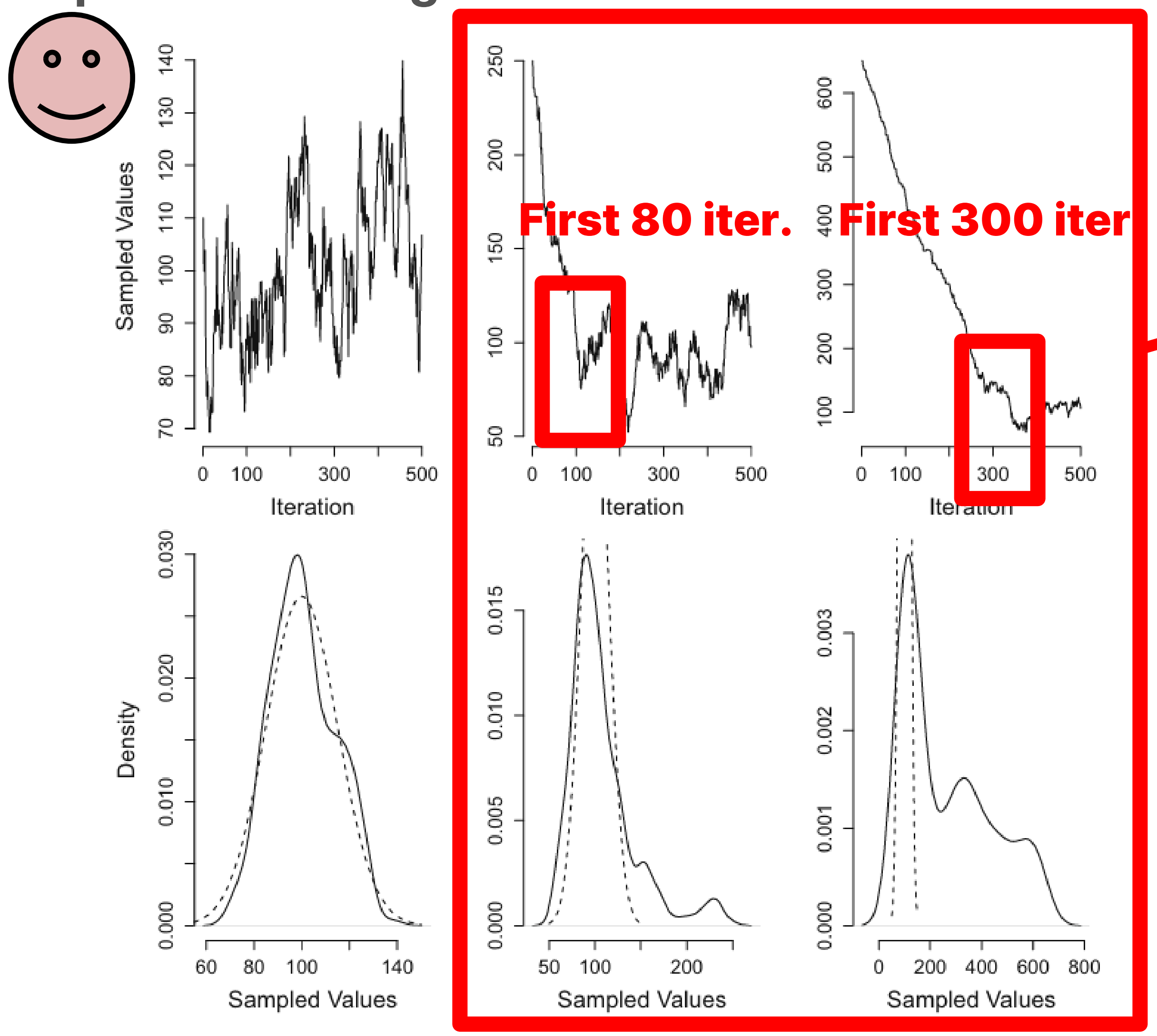


Fig. 1 A simple example of MCMC. *Left column:* A sampling chain starting from a good starting value, the mode of the true distribution. *Middle column:* A sampling chain starting from a starting value in the tails of the true distribution. *Right column:* A sampling chain starting from a value far from the true distribution. *Top row:* Markov chain. *Bottom row:* sample density. The analytical (true) distribution is indicated by the *dashed line*

Better Starting points | **Burn-in** | **Convergence**

- **WRONG initial value → incorrect reflection**
- **The first few iterations in any Markov chain cannot safely be assumed to be drawn from the target distribution. → Burn-in**

03 In-class example - limitation

Problem	Solution
Local maxima	
The practical performance of the samples depends on the value of the tuning parameter (== width of the proposal distribution)	Many augmented methods : Auto-tuning algorithms
Initial samples should be ignored as they might be very wrong.	• Convergence, burn-in, Better starting points (Fig 1) multiple chains with different starting values • Remove the early samples from non-stationary parts of the chain for convergence • Decisions about burn-in must be made after sampling. • Conversative / R statistic(Gelman & Rubin)

04

MCMC applied to a cognitive model

- Multivariate
- Gibbs-Sampling
- Tuning parameter

04 Cognitive model(SDT) -review

01

Goal : two parameters of SDT
(Signal Detection Theory) : d′, C

Target distribution
: Multivariate random variable

02

Gibbs Sampling

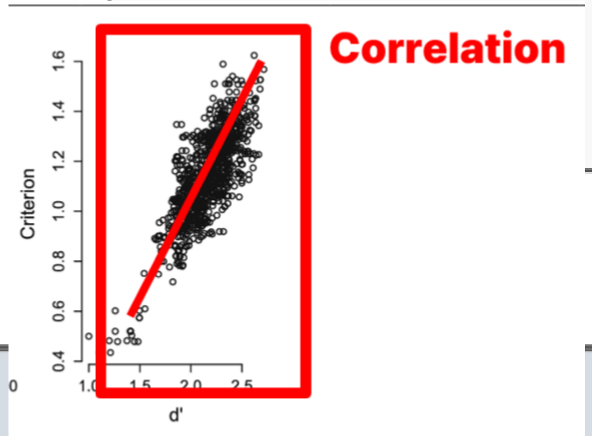
- multivariate

KEYWORD

03

Correlation between parameters

- The relative likelihood of parameter values of d′ will differ for different parameter values of C.



04

Blocking

- Separation of sampling between certain sets of parameters

04 Cognitive model(SDT)

< Gibbs Sampling >

Breaks down the problem by drawing for each parameter directly from that parameter's conditional distribution, or the probability distribution of a parameter given a specific value of another parameter.

Multivariate Density

Conditional distribution

Correlated

joint distribution of p variables through sampling

Example : p = 3

Let X=(X1,X2,X3) → we want to find this joint distribution, but sampling X =(X1, X2, X3) , directly is challenging.

Using conditional distribution makes random number generation easier → Gibbs Sampling!

→ $X_1 | X_2, X_3$

1) $X_1 | X_2, X_3$ → sampling a X_1^*, depending on X_2, X_3

$X_2 | X_1, X_3$ → sampling a X_2^*

$X_3 | X_1, X_2$ 〃 X_3^*

repeating!

sample!

→ after one iteration, we can get (X_1^*, X_2^*, X_3^*)

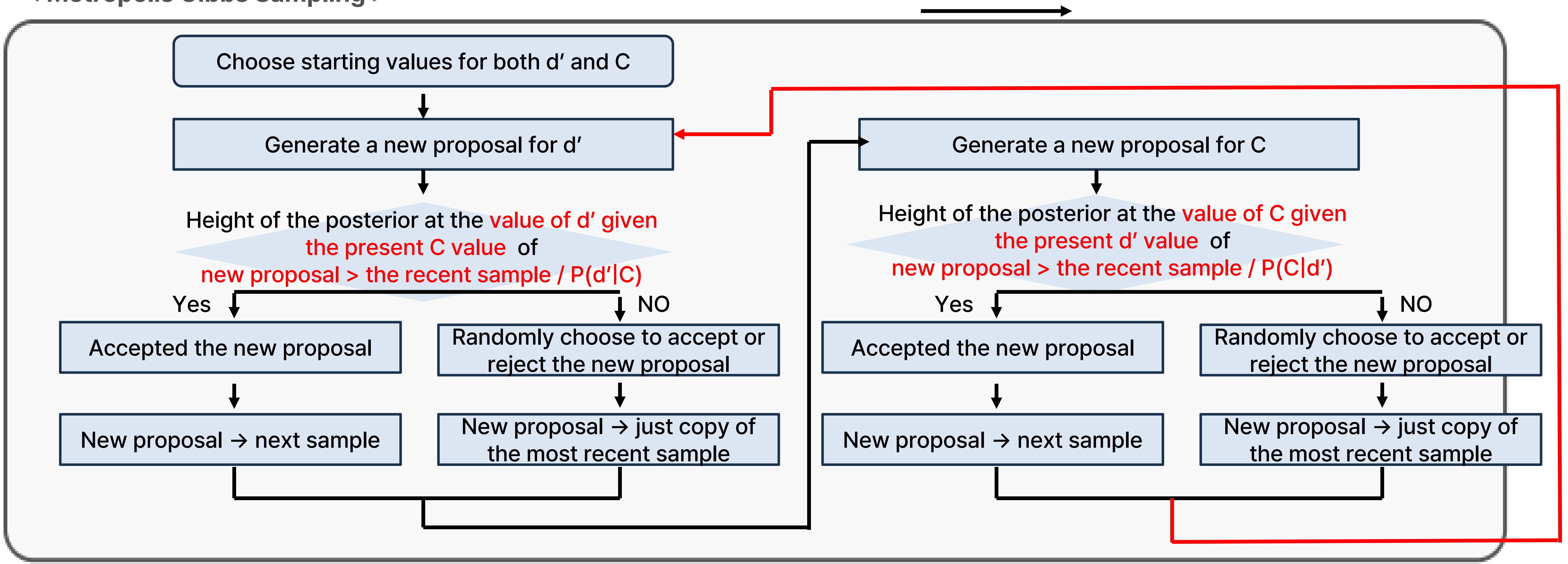
→ After this chain converges, it eventually stabilizes at a certain iteration.

ex)

* After convergence, the sample (X_1^*, X_2^*, X_3^*) will follow the joint distribution.

04 Cognitive model(SDT)

< Metropolis Gibbs Sampling >



04 Cognitive model(SDT)

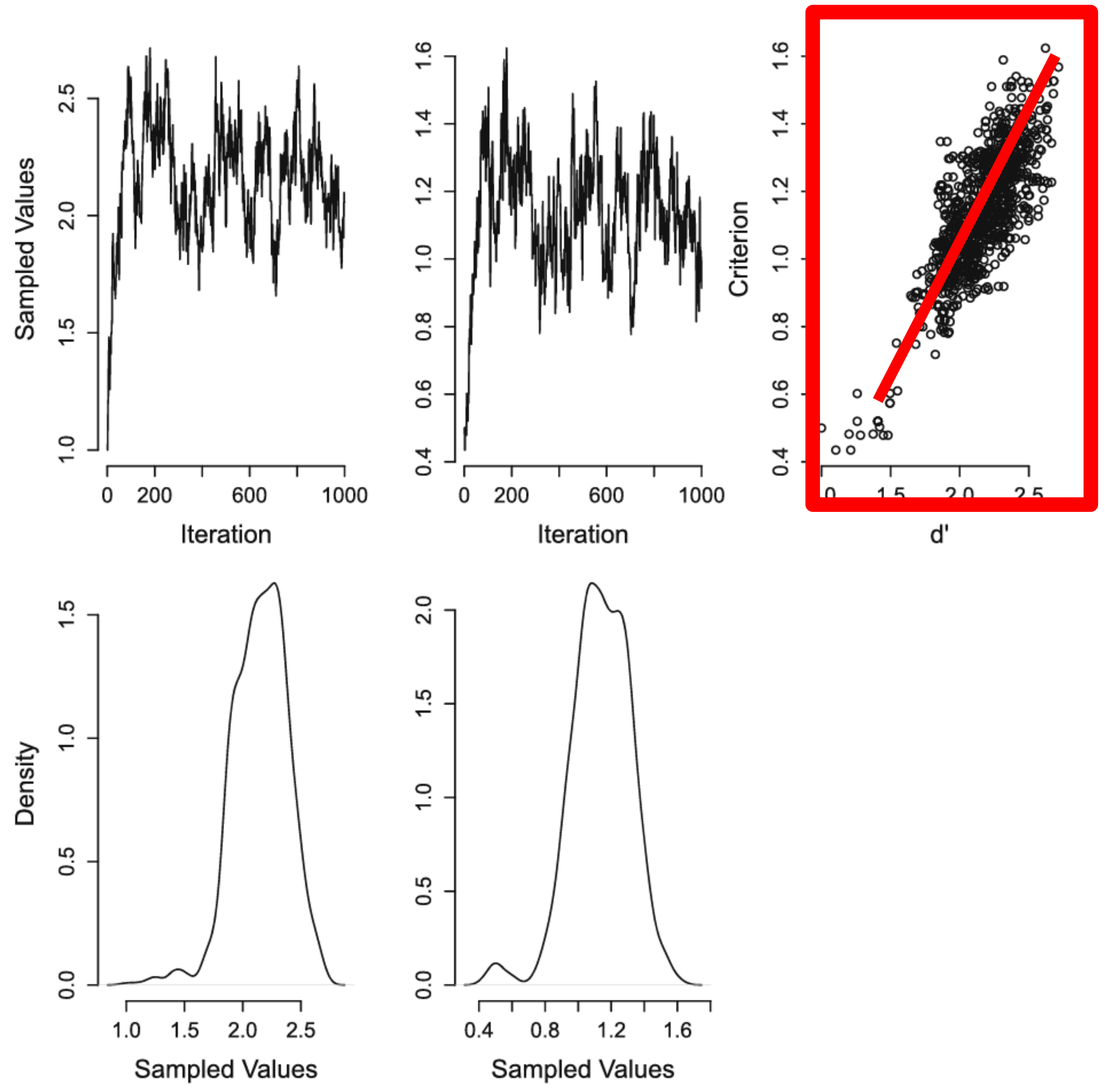


Fig. 2 An example of Metropolis within Gibbs sampling. *Left column:* Markov chain and sample density of d'. *Middle column:* Markov chain and sample density of C. *Right column:* The joint samples, which are clearly correlated

- **Samples out of the joint posterior, which is a bivariate distribution → We can find the correlation between parameters**

<Gibbs Sampling>

+) Cognitive model: correlation typical!!

- **with Metropolis: alleviate the problem that sampling proposals from <u>an uncorrelated joint distribution ignores that the probability distribution of each parameter differs</u> depending on the values of the other parameters.**
- **Problem: uncorrelated proposal distribution does not match the correlated target distribution.**

05

Sampling beyond basic metropolis-hastings

- Differential Evolution
- Gibbs-Sampling
- Uncorrelated joint distribution

05 Gibbs Sampling in DE

Differential Evolution(DE)

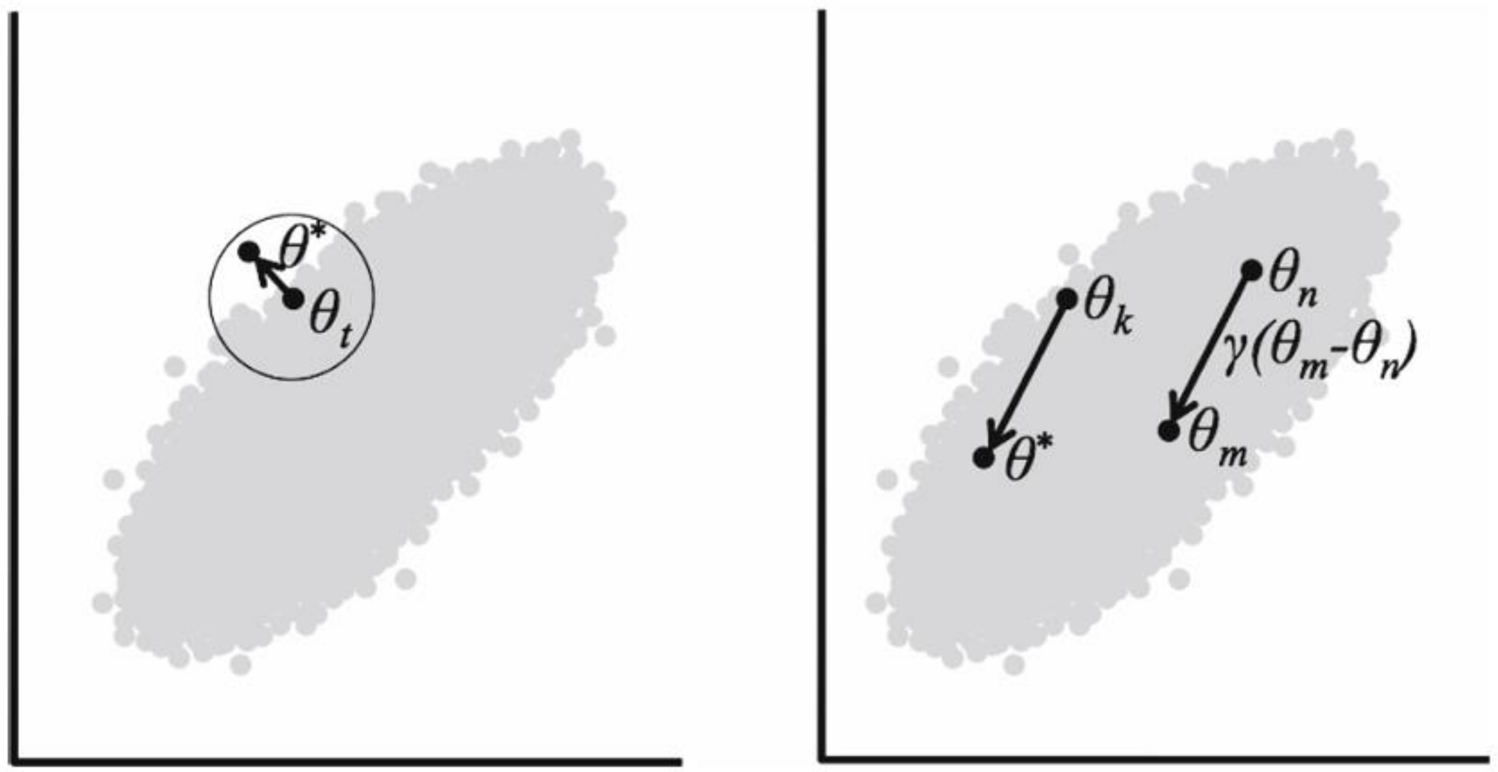


Fig. 3 *Left panel:* MCMC sampling using a conventional symmetrical proposal distribution. *Right panel:* MCMC sampling using the crossover method in Differential Evolution. See text for details

- **To improve proposals and have them respect the parameter correlation.**
- **Use multiple chains: with a set of many initial guesses**
- **Generate one chain of samples from each initial guess**
- **The chains are not independent – interact with each other during sampling**
- **Uses the difference between other chains to generate new proposal values**
- **Γ :tuning parameters (depending on the # of parameters)**

05 Gibbs Sampling in DE

Differential Evolution(DE)

1. To generate a proposal for the new value of chain θ_k, first choose two other chains at random. Suppose these are chains n and m. Find the distance between the current samples for those two chains, i.e.: $\theta_m - \theta_n$.
2. Multiply the distance between chains m and n by a value γ. Create the new proposal by adding this multiplied distance to the current sample. So, the proposal so far is: $\theta_k + \gamma(\theta_m - \theta_n)$. The value γ is a tuning parameter of the DE algorithm.
3. Add a very small amount of random noise to the resulting proposal, to avoid problems with identical samples ("degeneracy"). This leads to the new proposal value, θ^*.

- The default values(γ, the size of the "very small amount of random noise)
  - almost "auto-tuning" (ter Braak, 2006).
  - Typically, the random noise ~ Uniform (centered on zero and which is very narrow)
  - For the SDT example, where the d' and C parameters are in the region of 0.5–1, the random noise might be sampled from a uniform distribution with minimum-0.001 and maximum +0.001.
  - The γ parameter should be selected differently depending on the number of parameters in the model to be estimated, but a good guess is $2.38/\sqrt{(2K)}$,where K is the number of parameters in the model.

06

Summary

- Method : Metropolis (-Hastings), Gibbs Sampling, DE
- Tips : multiple chains, burn-in, tuning parameters
- Application

06 MCMC sampling

Metropolis	Metropolis-Hastings	Gibbs Sampling	Gibbs Sampling in DE (Differential Evolution)
Symmetric	**Asymmetric**	**Multivariate & uncorrelated joint distribution**	**Solution for mismatch between the target and proposal distribution**
Used when Univariate & Multivariate	Used when Univariate & Multivariate	Better able to capture correlated distributions of parameters sampling from conditional distributions problem: uncorrelated distribution doesn't match the correlated target distribution.	Starts with a set of many initial guesses, and generates one chain of samples from each initial guess. Considers the correlation in the distribution

ALL MCMC methods have "tuning parameters" that need to be adjusted to make the algorithm sample efficiently.

06 Application

Data Imputation()

Topic Modeling(LDA)

Classification ...

MICE

mice: Multivariate Imputation by Chained Equations in R

Stef van Buuren, Karin Groothuis-Oudshoorn

Abstract

The R package **mice** imputes incomplete multivariate data by chained equations. The software mice 1.0 appeared in the year 2000 as an S-PLUS library, and in 2001 as an R package. mice 1.0 introduced predictor selection, passive imputation and automatic pooling. This article documents mice, which extends the functionality of mice 1.0 in several ways. In **mice**, the analysis of imputed data is made completely general, whereas the range of models under which pooling works is substantially extended. **mice** adds new functionality for imputing multilevel data, automatic predictor selection, data handling, post-processing imputed values, specialized pooling routines, model selection tools, and diagnostic graphs. Imputation of categorical data is improved in order to bypass problems caused by perfect prediction. Special attention is paid to transformations, sum scores, indices and interactions using passive imputation, and to the proper setup of the predictor matrix. **mice** can be downloaded from the Comprehensive R Archive Network. This article provides a hands-on, stepwise approach to solve applied incomplete data problems.

https://www.jstatsoft.org/article/view/v045i03

ELSEVIER

Information Systems

Volume 94, December 2020, 101582

A review of topic modeling methods

Ike Vayansky [a], Sathish A.P. Kumar [b]

Show more

Outline | Share | Cite

https://doi.org/10.1016/j.is.2020.101582

Get right

Highlights

- Reviewed different topic modeling approaches dealing with correlation between topics.
- This review will encourage more diversity when performing topic modeling.
- The classification of methods in our review is flexible.
- Discussed the techniques of optimizing the topic modeling algorithms.
- Created and presented a decision tree to select a topic modeling method

https://www.sciencedirect.com/science/article/pii/S0306437920300703?casa_token=jOyoCo4IMjEAAAAA:IM_q8sYBrB0mPMbvoTEn4Qd8-dcvORuKo35HJnyoJt-tQgJjLUYG_q0fCXD-ntAONcbNIo-Ped8_

ELSEVIER

Computational Statistics & Data Analysis

Volume 51, Issue 7, 1 April 2007, Pages 3529-3550

Interpretation and inference in mixture models: Simple MCMC works

John Geweke

Show more

Outline | Share | Cite

https://doi.org/10.1016/j.csda.2006.11.026

Get rights and content

Abstract

The mixture model likelihood function is invariant with respect to permutation of the components of the mixture. If functions of interest are permutation sensitive, as in classification applications, then interpretation of the likelihood function requires valid inequality constraints and a very large sample may be required to resolve ambiguities. If functions of interest are permutation invariant, as in prediction applications, then there are no such problems of interpretation. Contrary to assessments in some recent publications, simple and widely used Markov chain Monte Carlo (MCMC) algorithms with data augmentation reliably recover the entire posterior distribution.

https://www.sciencedirect.com/science/article/pii/S0167947306004506?casa_token=TlqGfXMV9fAAAAAA:4c1-I2lmRrrgMstE92dC79vDQSqfVCzjxmszTHXfbGGM3zMw4qODTuZM5QDKiqI9qzyU_wgDqyyo