Towards More Efficient MCMC Sampling

Presenter: Quan Zhou

Department of Statistics, Texas A&M University

Markov chain Monte Carlo sampling

Markov chain Monte Carlo (MCMC) algorithms can generate samples from a target distribution π by simulating a Markov chain with *stationary* distribution π.

Example: Metropolis-Hastings (MH) algorithms, Gibbs samplers.

Markov chain Monte Carlo sampling

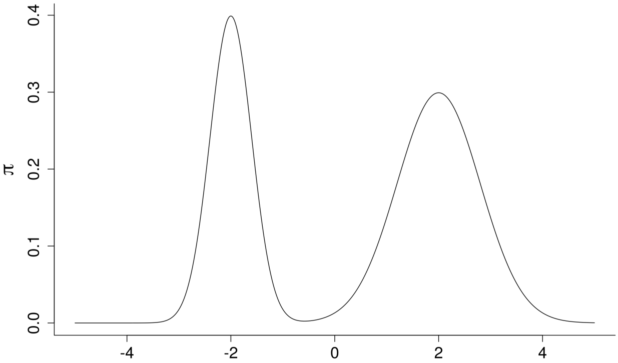
Markov chain Monte Carlo (MCMC) algorithms can generate samples from a target distribution π by simulating a Markov chain with *stationary* distribution π.

Example: Metropolis-Hastings (MH) algorithms, Gibbs samplers.

Widely used in Bayesian statistics, since posterior distributions often involve intractable normalizing constants.

Markov chain Monte Carlo sampling

Sampling and optimization are closely related: Dalalyan [2017a], Ma et al. [2019], Talwar [2019].



Example 1: variable selection

Consider the linear regression model

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}, \boldsymbol{\epsilon} \in \mathbb{R}^n$, $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\boldsymbol{Z}$ is an $n \times p$ design matrix. We assume most entries of $\boldsymbol{\beta}$ are zero, and our goal is to identify

$$\gamma = \{1 \leq k \leq p \colon \beta_p \neq 0\}.$$

Example 1: variable selection

Consider the linear regression model

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}, \boldsymbol{\epsilon} \in \mathbb{R}^n$, $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\boldsymbol{Z}$ is an $n \times p$ design matrix. We assume most entries of $\boldsymbol{\beta}$ are zero, and our goal is to identify

$$\gamma = \{1 \leq k \leq p \colon \beta_p \neq 0\}.$$

Search space

The search space is $2^{\{1,\ldots,p\}}$, which has cardinality 2^p.

Example 1: variable selection

In high-dimensional settings, sparsity constraints need to be imposed, but usually the search space still grows *super-polynomially* in p.

Example 1: variable selection

In high-dimensional settings, sparsity constraints need to be imposed, but usually the search space still grows *super-polynomially* in p.

Local algorithms

Most sampling algorithms for variable selection are “local”: the next move is selected from a “small” set of neighboring states which has cardinality *polynomial* in p.

Example 1: variable selection

In high-dimensional settings, sparsity constraints need to be imposed, but usually the search space still grows *super-polynomially* in p.

Local algorithms

Most sampling algorithms for variable selection are "local": the next move is selected from a "small" set of neighboring states which has cardinality *polynomial* in p.

Example: a typical search path in variable selection.

$$\{1,2\} \xrightarrow{\text{add covariate 4}} \{1,2,4\} \xrightarrow{\text{swap covariate 2 with 3}} \{1,3,4\}$$

$$\xrightarrow{\text{delete covariate 4}} \{1,3\} \xrightarrow{\text{delete covariate 1}} \{3\}$$

Example 1: variable selection

Heritability estimation

In addition to variable selection, we also want to estimate $\mathrm{Var}(\boldsymbol{\epsilon})/\mathrm{Var}(\boldsymbol{y})$.

Example 2: structure learning

DAG model

A p-variate directed acyclic graph (DAG) encodes the conditional independence (CI) relations among p node variables.

Structure learning

Learn the underlying DAG model of a p-variate probability distribution from n i.i.d. observations, $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n$; each $\boldsymbol{Z}_i \in \mathbb{R}^p$.

Example 2: structure learning

DAG model

A p-variate directed acyclic graph (DAG) encodes the conditional independence (CI) relations among p node variables.

Structure learning

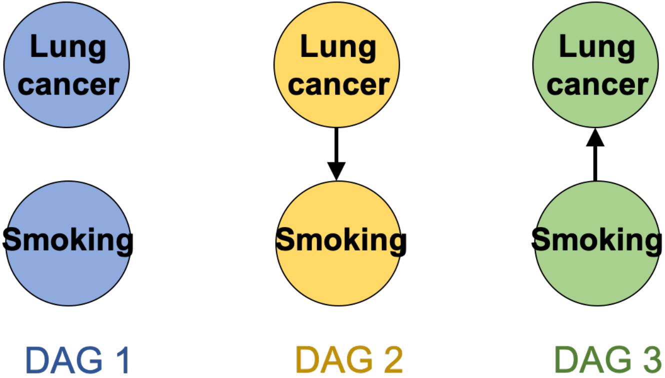
Learn the underlying DAG model of a p-variate probability distribution from n i.i.d. observations, $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n$; each $\boldsymbol{Z}_i \in \mathbb{R}^p$.

Search space

The collection of all p-vertex labeled DAGs; cardinality is super-exponential in p.

Example 2: structure learning

For two variables, there are 3 possible DAGs.



Example 3: estimation of PDE parameters

Suppose that we have i.i.d. observations $(z_1, y_1), (z_2, y_2), \dots, (z_n, y_n)$ generated from

$$y_i = f(z_i; \boldsymbol{\theta}) + \epsilon_i,$$

where f is the solution to a partial differential equation (PDE) parameterized by $\boldsymbol{\theta}$. Our goal is to estimate $\boldsymbol{\theta} \in \mathbb{R}^p$.

Example 3: estimation of PDE parameters

Suppose that we have i.i.d. observations $(z_1, y_1), (z_2, y_2), \ldots, (z_n, y_n)$ generated from

$$y_i = f(z_i; \boldsymbol{\theta}) + \epsilon_i,$$

where f is the solution to a partial differential equation (PDE) parameterized by $\boldsymbol{\theta}$. Our goal is to estimate $\boldsymbol{\theta} \in \mathbb{R}^p$.

Search space

The parameter space $\mathbb{R}^p$. Though it is continuous, gradient-based sampling methods cannot be applied.

Metropolis-Hastings (MH) algorithms

Let $\mathcal{X}$ be a finite state space on which each x has N neighbors. We write $y \sim x$ if y is a neighbor of x; assume $x \sim y$ whenever $y \sim x$.

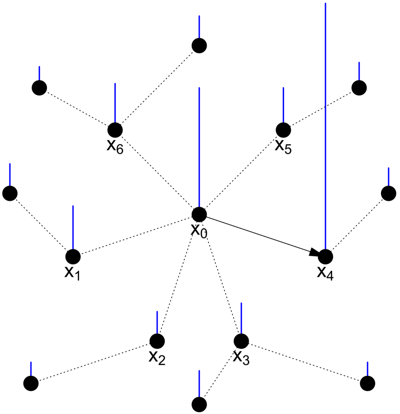
Metropolis-Hastings (MH) algorithms

Let $\mathcal{X}$ be a finite state space on which each x has N neighbors. We write $y \sim x$ if y is a neighbor of x; assume $x \sim y$ whenever $y \sim x$.

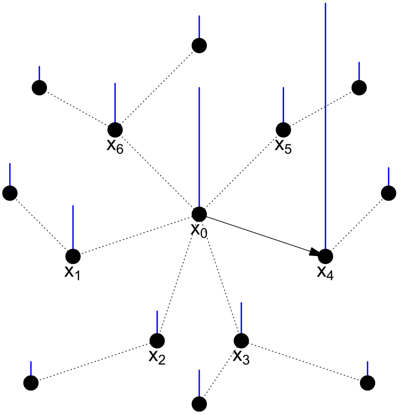
Random walk MH algorithm targeting π

An iteration at state x:

1. Draw a neighbor y randomly with equal probability.
2. Accept y with probability
$$a(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}.$$
3. If y is accepted, we move to y; otherwise, stay at x.



Each dot represents a state, and the height of the blue bar indicates $\pi(\cdot)$.
At point x_0, the best move is $x_0 \to x_4$.



At point x_0, a random walk proposal proposes x_4 with probability $1/6$.
We may use a locally informed proposal to increase this probability.

Informed MH

Choose a non-decreasing function $h\colon (0,\infty) \to (0,\infty)$.

Locally informed MH algorithm targeting π

An iteration at state x:

1. Draw $y \sim x$ with probability proportional to $h(\pi(y)/\pi(x))$.
2. Accept y with probability

$$a_h(x,y) = \min\left\{1, \quad \frac{\pi(y)h\left(\frac{\pi(x)}{\pi(y)}\right) Z_h(x)}{\pi(x)h\left(\frac{\pi(y)}{\pi(x)}\right) Z_h(y)}\right\},$$

$$\text{where } Z_h(x) = \sum_{x' \colon x' \sim x} h\left(\frac{\pi(x')}{\pi(x)}\right).$$

3. If y is accepted, we move to y; otherwise, stay at x.

Remarks on informed proposals

- Similar ideas are used in many MCMC methods [Titsias and Yau, 2017, Zanella and Roberts, 2019, Zanella, 2020, Griffin et al., 2021] and some non-MCMC methods [Hans et al., 2007, Shin et al., 2018].
- To implement an informed proposal at x, we need to evaluate $\pi(y)$ for each $y \sim x$; this can be parallelized.
- Difficult to control the acceptance probability.
- Informed MH algorithms can mix even more slowly than RWMH.

Question 1: Do we have theoretical guarantees?

Question 1: Do we have theoretical guarantees?

Define mixing time by $T_{\text{mix}} = \max_x \min\{t \colon ||P^t(x, \cdot) - \pi||_{\text{TV}} \le 1/4\}$.

Under a "unimodal condition" on π (precise statements given later),

- For random walk MH, the mixing time is $O(N \log \pi_{\min}^{-1})$ where
  - $\pi_{\min} = \min_{x \in \mathcal{X}} \pi(x)$,
  - N is the neighborhood cardinality.

Question 1: Do we have theoretical guarantees?

Define mixing time by $T_{\text{mix}} = \max_x \min\{t\colon ||P^t(x, \cdot) - \pi||_{\text{TV}} \le 1/4\}$.

Under a "unimodal condition" on π (precise statements given later),

- For random walk MH, the mixing time is $O(N \log \pi_{\min}^{-1})$ where
  - $\pi_{\min} = \min_{x \in \mathcal{X}} \pi(x)$,
  - N is the neighborhood cardinality.
- There exists an informed MH with mixing time $O(\log \pi_{\min}^{-1})$.

See Zhou et al. [2022], Zhou and Chang [2023] for general results and the analysis of variable selection and structure learning.

Question 2: Do we have to use MH schemes?

Question 2: Do we have to use MH schemes?

Recall informed proposals draw $y \sim x$ with probability $\propto h(\pi(y)/\pi(x))$.
Now we assume h is a balancing function.

Balancing function [Zanella, 2020]

We say $h\colon (0,\infty) \to (0,\infty)$ is a *balancing* function if

$$h(u) = u\,h(1/u), \quad \forall\, u > 0.$$

Examples: $h(u) = 1 + u, h(u) = \min\{1, u\}, h(u) = \sqrt{u}$.

Question 2: Do we have to use MH schemes?

Recall informed proposals draw $y \sim x$ with probability $\propto h(\pi(y)/\pi(x))$. Now we assume h is a balancing function.

Balancing function [Zanella, 2020]

We say $h\colon (0,\infty) \to (0,\infty)$ is a *balancing* function if

$$h(u) = u\, h(1/u), \quad \forall\, u > 0.$$

Examples: $h(u) = 1 + u, h(u) = \min\{1, u\}, h(u) = \sqrt{u}$.

Our solution is very simple: always accept the informed proposal and use importance sampling to correct for the bias.

Question 2: Do we have to use MH schemes?

Informed importance tempering (IIT)

Choose a balancing function h. An iteration at state x:

1. Calculate $h(\pi(y)/\pi(x))$ for every $y \sim x$.
2. Calculate $Z_h(x) = \sum_{y \sim x} h(\pi(y)/\pi(x))$.
3. Assign to x importance weight $1/Z_h(x)$.
4. Move to x_{next} with probability proportional to $h(\pi(x_{\text{next}})/\pi(x))$.

Question 2: Do we have to use MH schemes?

Informed importance tempering (IIT)

Choose a balancing function h. An iteration at state x:

1. Calculate $h(\pi(y)/\pi(x))$ for every $y \sim x$.
2. Calculate $Z_h(x) = \sum_{y \sim x} h(\pi(y)/\pi(x))$.
3. Assign to x importance weight $1/Z_h(x)$.
4. Move to x_{next} with probability proportional to $h(\pi(x_{\text{next}})/\pi(x))$.

This generalizes the tempered Gibbs sampler of Zanella and Roberts [2019], an MCMC scheme for variable selection that can be seen as IIT with balancing function $h(u) = 1 + u$.

Question 3: How to choose the informed proposal?

Question 3: How to choose the informed proposal?

In Zhou and Smith [2022], we show that:

- IIT with $h(u) = 1 + u$ converges "extremely fast" (see our paper for definition).

Question 3: How to choose the informed proposal?

In Zhou and Smith [2022], we show that:

- IIT with $h(u) = 1 + u$ converges "extremely fast" (see our paper for definition).
- However, $h(u) = 1 + u$ is too aggressive and can be very inefficient for multimodal targets.

Question 3: How to choose the informed proposal?

In Zhou and Smith [2022], we show that:

- IIT with $h(u) = 1 + u$ converges "extremely fast" (see our paper for definition).
- However, $h(u) = 1 + u$ is too aggressive and can be very inefficient for multimodal targets.
- $h(u) = \sqrt{u}$ performs well in a wider range of settings.

Question 4: Is importance tempering a general technique?

Question 4: Is importance tempering a general technique?

In Li et al. [2023], we propose the following IIT variants:

- IIT schemes that do not require posterior evaluation of all neighbors;
- integration of IIT and simulated tempering algorithm;
- integration of IIT and pseudo-marginal methods;
- importance-tempered multiple-try algorithm, which is applicable to general state spaces.

Question 4: Is importance tempering a general technique?

In Li et al. [2023], we propose the following IIT variants:

- IIT schemes that do not require posterior evaluation of all neighbors;
- integration of IIT and simulated tempering algorithm;
- integration of IIT and pseudo-marginal methods;
- importance-tempered multiple-try algorithm, which is applicable to general state spaces.

IIT schemes appear to always converge faster than their MH counterparts in our numerical studies.

Future research projects

1. Developing importance tempering-based MCMC algorithms for variable selection, graphical models, PDE learning, heritability estimation, stochastic neural networks, etc. Potential challenges:

- realistic, flexible and hierarchical Bayesian modeling
- efficient, high-quality implementation
- approximating informed proposals
- more sophisticated MCMC schemes
- experience and knowledge about competing methods
- handling complex real data
- interdisciplinary knowledge

Future research projects

2. Online estimation of MCMC convergence, especially for problems with finite state spaces.

- Mostly computational, but knowledge about Markov chain mixing will be useful.
- Methodology for IIT samplers need to be further developed.

Future research projects

2. Online estimation of MCMC convergence, especially for problems with finite state spaces.

- Mostly computational, but knowledge about Markov chain mixing will be useful.
- Methodology for IIT samplers need to be further developed.

3. Adaptive MCMC methods for multimodal targets.

- How to choose state space partition?
- How to allocate computational budget?
- How to learn the optimal temperature?
- How to learn the optimal informed proposal scheme?

Skills to learn

- Linear algebra, especially numerical linear algebra
- Programming: Cpp, python, etc.
- Statistical simulation
- MCMC theory and methodology
- High-dimensional theory and methodology for Bayesian statistics

Thank you!

- Q. Zhou and H. Chang. “Complexity analysis of Bayesian learning of high-dimensional DAG models and their equivalence classes.” *Annals of Statistics*, `arXiv:2101.04084`.
- Q. Zhou, J. Yang, D. Vats, G. Roberts and J. Rosenthal. “Dimension-free mixing of high-dimensional Bayesian variable selection.” *JRSSB*, `arXiv:2105.05719`.
- Q. Zhou and A. Smith. “Rapid convergence of informed importance tempering.” *AISTATS* (oral presentation), `arXiv:2107.10827`.
- G. Li, A. Smith and Q. Zhou. “Importance is Important: A Guide to Informed Importance Tempering Methods.” `arXiv:2304.06251`.

References I

Hyunwoong Chang, Changwoo Lee, Zhao Tang Luo, Huiyan Sang, and Quan Zhou. Rapidly mixing multiple-try Metropolis algorithms for model selection problems. *Advances in Neural Information Processing Systems*, 35:25842–25855, 2022.

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Conference on learning theory*, pages 300–323. PMLR, 2018.

Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent. In *Conference on Learning Theory*, pages 678–689. PMLR, 2017a.

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):651–676, 2017b.

Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast! In *Conference on learning theory*, pages 793–797. PMLR, 2018.

Philippe Gagnon, Florian Maire, and Giacomo Zanella. Improving multiple-try Metropolis with local balancing. *arXiv preprint arXiv:2211.11613*, 2022.

Robert Gramacy, Richard Samworth, and Ruth King. Importance tempering. *Statistics and Computing*, 20:1–7, 2010.

References II

JE Griffin, KG Łatuszyński, and MFJ Steel. In search of lost mixing time: adaptive Markov chain Monte Carlo schemes for Bayesian variable selection with very large p. *Biometrika*, 108(1):53–69, 2021.

Chris Hans, Adrian Dobra, and Mike West. Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007.

Guanxun Li, Aaron Smith, and Quan Zhou. Importance is important: A guide to informed importance tempering methods. *arXiv preprint arXiv:2304.06251*, 2023.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, 2019.

Oren Mangoubi and Aaron Smith. Mixing of Hamiltonian Monte Carlo on strongly log-concave distributions 2: Numerical integrators. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 586–595. PMLR, 2019.

Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. *arXiv preprint arXiv:1909.05503*, 2019.

Minsuk Shin, Anirban Bhattacharya, and Valen E Johnson. Scalable Bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. *Statistica Sinica*, 28(2):1053, 2018.

References III

Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, probability and Computing*, 1(4):351–370, 1992.

Kunal Talwar. Computational separations between sampling and optimization. *Advances in neural information processing systems*, 32, 2019.

Michalis K Titsias and Christopher Yau. The Hamming ball sampler. *Journal of the American Statistical Association*, 112(520):1598–1611, 2017.

Xiaodong Yang and Jun S Liu. Convergence rate of multiple-try Metropolis independent sampler. *Statistics and Computing*, 33(4):79, 2023.

Yun Yang, Martin J Wainwright, and Michael I Jordan. On the computational complexity of high-dimensional Bayesian variable selection. *The Annals of Statistics*, 44(6):2497–2532, 2016.

Giacomo Zanella. Informed proposals for local MCMC in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, 2020.

Giacomo Zanella and Gareth Roberts. Scalable importance tempering and Bayesian variable selection. *Journal of the Royal Statistical Society Series B*, 81(3):489–517, 2019.

Quan Zhou and Hyunwoong Chang. Complexity analysis of Bayesian learning of high-dimensional DAG models and their equivalence classes. *Annals of Statistics*, to appear, 2023.

References IV

Quan Zhou and Aaron Smith. Rapid convergence of informed importance tempering. pages 10939–10965, 2022.

Quan Zhou, Jun Yang, Dootika Vats, Gareth O Roberts, and Jeffrey S Rosenthal. Dimension-free mixing for high-dimensional bayesian variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(5):1751–1784, 2022.

Bumeng Zhuo and Chao Gao. Mixing time of Metropolis-Hastings for Bayesian community detection. *Journal of Machine Learning Research*, 22:10–1, 2021.