STA496: Examining the efficiency of Markov chain Monte Carlo algorithms

Tianye Dou, Aidan Li, Liyan Wang

University of Toronto

Monte Carlo Methods

Monte Carlo methods rely on repeated random sampling and statistical analysis to compute certain quantities.
The goal of Monte Carlo methods:

- To generate samples $\{x^{(i)}\}_{i=1}^{N}$ from a given target probability distribution with density $\pi(x)$.
- To estimate expectations of functions under this distribution (using integration).

Motivation

The target density $\pi(x)$ can be complicated and high-dimensional, making it hard to sample from.
It is difficult to evaluate $\pi(x)$ everywhere, especially if $\pi(x)$ has high dimensionality.
So how should we sample from $\pi(x)$?

Markov Chain Monte Carlo (MCMC)

Markov Chain: A Markov chain is a probabilistic model that characterizes a series of potential events, where the likelihood of each event is determined solely by the state of the preceding event. In other words, we assume nothing other than the state of time i affects the state of time $i + 1$.

Markov Chain Monte Carlo

- Given a complicated and high dimensional target distribution with density $\pi(\cdot)$
- Use Markov chain to generate a sequence of x values, denoted as $x_0, x_1, x_2, ...$, which converges to distribution $\pi(\cdot)$
- For n sufficiently large, $x_n, x_{n+1}, ...$ are samples from $\pi(\cdot)$.
  i.e. If we run the chain long enough, it gives us samples from $\pi(\cdot)$.

The Metropolis-Hastings algorithm

Suppose we have a target distribution with density π and the current state of the Markov chain is x_n. We generate x_{n+1} from a three-step process.

1. The proposal step: Sample a candidate x^* from the proposal density $q(x^*|x_n)$.
2. The accept-reject step: Accept the new state with probability $a = min(1, \frac{\pi(x^*)}{\pi(x_n)} \frac{q(x_n|x^*)}{q(x^*|x_n)})$.
3. If we accept the new state, $x_{n+1} = x^*$. If we reject the new state, $x_{n+1} = x_n$.

Metropolis vs Metropolis-Hastings

Metropolis vs Metropolis-Hastings

The Metropolis algorithm assumes $q(x_n|x^*) = q(x^*|x_n)$, thus the acceptance probability a simplifies to $a = \frac{\pi(x^*)}{\pi(x_n)}$, i.e. just comparing the target density at the two points.

Metropolis Visualization

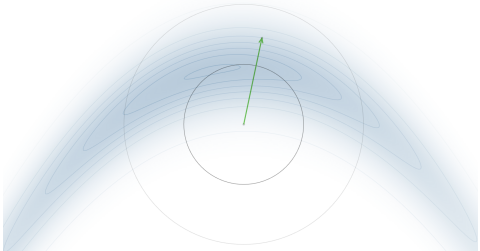


Figure: State 1

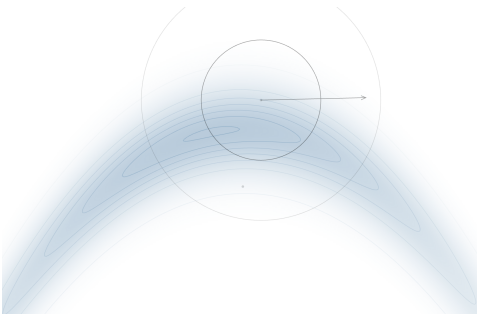
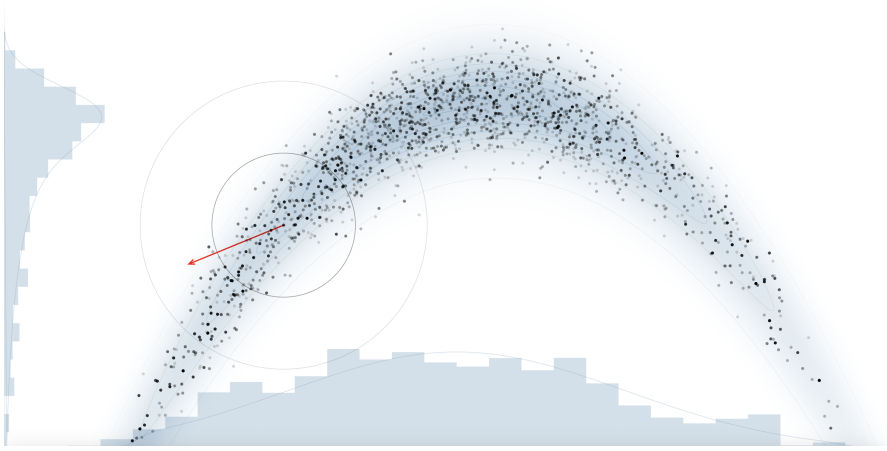


Figure: State 2

Visualization from `https://chi-feng.github.io/mcmc-demo/app.html`

Metropolis Visualization

After many iterations...



Efficiency and the Optimal Scaling Problem

The choice of proposal distribution, typically $N(x, \sigma^2)$ centred at x, is crucial for algorithm efficiency. Different "scaling", e.g. variance σ^2, gives different results.

- If σ is too small,
  1. Usually accept proposed steps, but the Markov chain won't explore much $\implies$ slow convergence to $\pi(x)$.
  2. Only one mode of the target density might be visited in a finite number of steps.
- If σ is too large,
  1. Usually reject proposed steps $\implies$ very slow to make a new exploration step. Slow speed of convergence to $\pi(x)$.
  2. May have high correlations in a finite number of steps.

Evaluating the Efficiency of MCMC Algorithms

Suppose our proposal is $Q = N(x, \frac{\ell^2}{d} I_d)$ where $\ell > 0$ is a scaling factor.

- In high-dimensions $d \to \infty$, we want to maximize the limiting speed $h(\ell)$.
- Acceptance rate $A(\ell) = \lim_{n \to \infty} \frac{\text{accepted moves}}{n}$ relates to limiting speed.
- Under strong assumptions, the limiting speed is maximized at $A \approx 0.234$.
- Optimize your MCMC algorithm by adjusting the proposal scaling so that the acceptance rate ≈ 0.234.
- Set your proposal to be $N(x, (2.38^2/d)\Sigma_*)$.

Note:

Even if the theorem's strong assumptions are not satisfied, 0.234 is still nearly optimal for many cases. An acceptance rate of 0.20-0.50 is still quite efficient.

Evaluating the Efficiency of MCMC Algorithms

The theory is for dimension $d \to \infty$, but what about lower dimensions?

- Even for $d \geq 5$ the 0.234 acceptance rate theory still seems to work pretty well!
- Also use Expected Squared Jumping Distance: $E[||x_{t+1} - x_t||^2]$. This metric measures how far, on average, the MCMC chain moves in a single iteration.
- Maximizing ESJD minimizes the first-order auto-correlation of the Markov chain, maximizing the number of "effective" samples and thus maximizing the efficiency.

Experiments with the Normal: Covariance Matrix with I.I.D. components

We start with a simple example. Assume dimension $d = 30$. Target distribution has density of form $\prod_{i=1}^{n} f(x_i)$, where $f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{x^2}{2}\right)$, and consider a proposal distribution $Q(x) = N(0, \sigma^2 I_d)$. Setting $\sigma^2 = 0.005$:

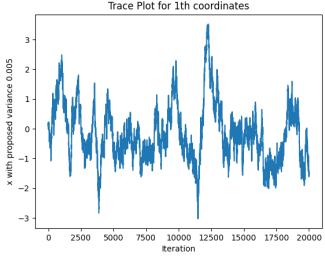


Figure: Trace plot of 1st coordinate with $\sigma^2 = 0.005$

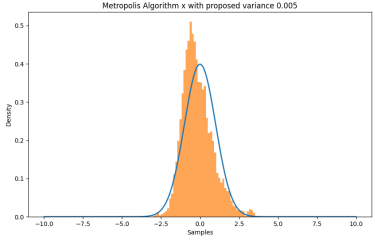


Figure: Histogram of 1st coordinate with $\sigma^2 = 0.005$

Experiments with the Normal: Covariance Matrix With I.I.D. components

We set a grid for the proposed variance σ^2 and obtained a corresponding plot of the relationship between ESJD and acceptance rate.

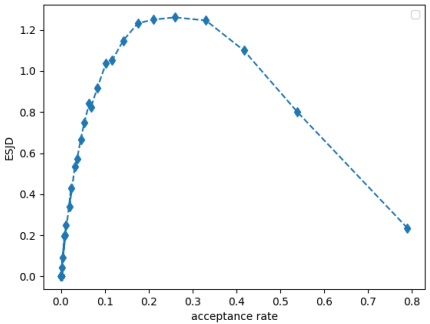


Figure: ESJD for Metropolis Algorithm as a function of acceptance rate

Experiments with the Normal: Covariance Matrix With I.I.D. components

With a few experiments, the proposed variance that maximizes ESJD with an acceptance rate of approximately 0.234 is 0.2.
With $\sigma^2 = 0.2$, and we could see that $0.2 \approx 2.38^2/d$

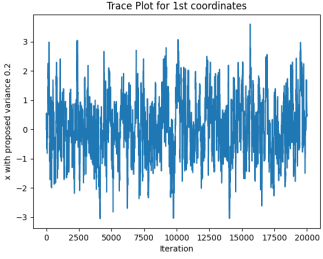


Figure: Trace plot of 1st coordinate with $\sigma^2 = 0.2$

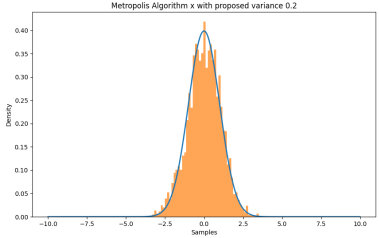


Figure: Histogram of 1st coordinate with $\sigma^2 = 0.2$

Starting Points

For IID components, the Markov chain starting point does not affect the convergence. In terms of dimensions from 10 to 100, efficiency is maximized when the acceptance rate is approximately 0.234.

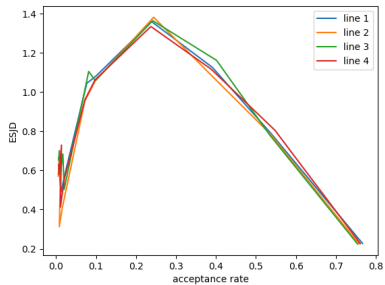


Figure: different starting points for IID covariance matrix

Dimensions

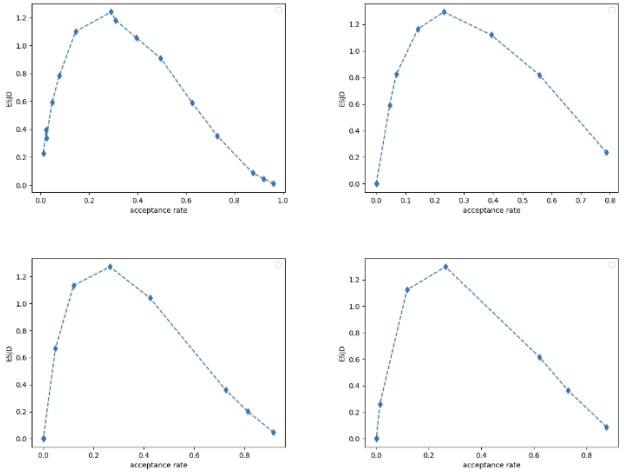


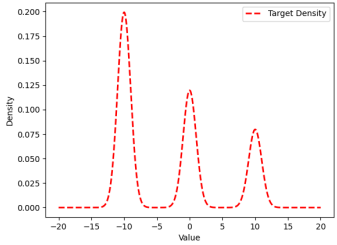
Figure: ESJD of IID covariance matrix as a function of acceptance rate, with dimension = 10 (top left), 30 (top right), 50 (bottom left), 100 (bottom right)

Multimodal Example

Assume dimension $d = 50$. Our target density is a product of IID components $\prod_{i=1}^{n} f(x_i)$. f is a one-dimensional density with three modes. Let $m_1, m_2, m_3 \in \mathbb{R}$. f is a mixture of Normals:

$$f = 0.5N(m_1, 1) + 0.3N(m_2, 1) + 0.2N(m_3, 1)$$

Here is the visualization of the first component:



Multimodal Example

With a normal proposal distribution $Q = N(x, \sigma^2 I_d)$, having an acceptance rate of 0.234 and a maximized ESJD no longer guarantees an "optimal" sampling.

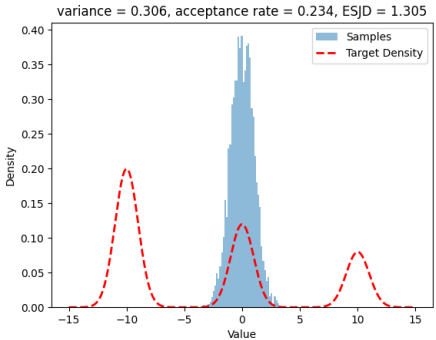


Figure: Histogram of the first coordinate

Multimodal Example

Parallel Tempering helps!

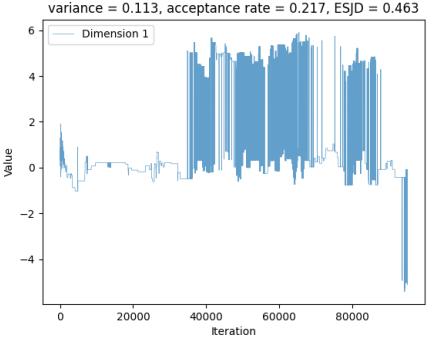


Figure: Traceplot of the first coordinate with parallel tempering

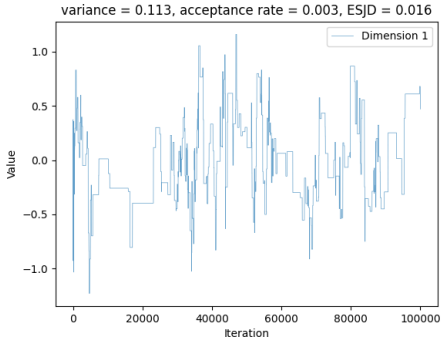


Figure: Traceplot of the first coordinate with standard RWM

Acknowledgements

- Professor Jeffrey Rosenthal, our supervisor, for this opportunity and his excellent guidance and support over the last semester in learning about and experimenting with a fascinating new topic.
- Austin Brown for his practical advice on optimizing our algorithm implementations and general directions for investigation.
- Piotr Zwiernik for giving helpful advice about this presentation and for allowing us to use this template.