

Modified Metropolis-Hastings algorithm with Delayed Rejection

Konstantin Zuev & Lambros Katafygiotis

Department of Civil Engineering
Hong Kong University of Science and Technology

zuev@ust.hk, lambros@ust.hk

19 June, 2008

Outline

- Introduction
- Modifications of the Metropolis-Hastings algorithm
 - ▶ Standard Metropolis-Hastings algorithm
 - ▶ Modified Metropolis-Hastings algorithm
 - ▶ Metropolis-Hastings algorithm with Delayed Rejection
 - ▶ Modified Metropolis-Hastings algorithm with Delayed Rejection
- Example
- Summary

Introduction

Problem: To find the probability of failure p_F

$$p_F = P(x \in F) = \int_{\mathbb{R}^N} I_F(x) \pi(x) dx$$

Introduction

Problem: To find the probability of failure p_F

$$p_F = P(x \in F) = \int_{\mathbb{R}^N} I_F(x) \pi(x) dx$$

Notation

- $x \in \mathbb{R}^N$ is a random vector
- π is the joint PDF of x
- $F \subset \mathbb{R}^N$ is a failure domain
- $I_F(x) = \begin{cases} 1, & x \in F; \\ 0, & x \notin F. \end{cases}$

Introduction

Problem: To find the probability of failure p_F

$$p_F = P(x \in F) = \int_{\mathbb{R}^N} I_F(x) \pi(x) dx$$

Notation

- $x \in \mathbb{R}^N$ is a random vector
- π is the joint PDF of x
- $F \subset \mathbb{R}^N$ is a failure domain
- $I_F(x) = \begin{cases} 1, & x \in F; \\ 0, & x \notin F. \end{cases}$

Assumptions

- N is very large, $N \sim 1000$
- p_F is very small, $p_F \sim 10^{-3} - 10^{-6}$
- π is the standard multidimensional Gaussian, $\pi = \mathcal{N}(0, I)$

Introduction

Problem: To find the probability of failure p_F

$$p_F = P(x \in F) = \int_{\mathbb{R}^N} I_F(x) \pi(x) dx$$

Notation

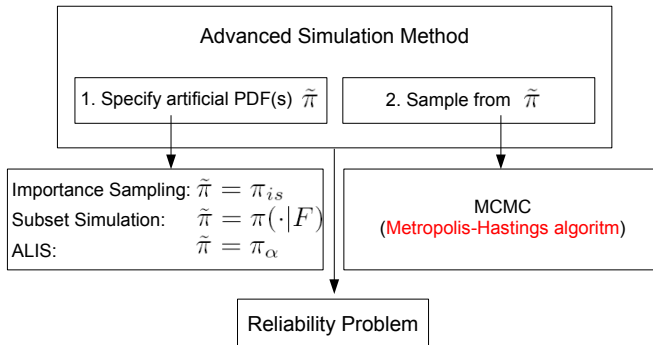
- $x \in \mathbb{R}^N$ is a random vector
- π is the joint PDF of x
- $F \subset \mathbb{R}^N$ is a failure domain
- $I_F(x) = \begin{cases} 1, & x \in F; \\ 0, & x \notin F. \end{cases}$

Assumptions

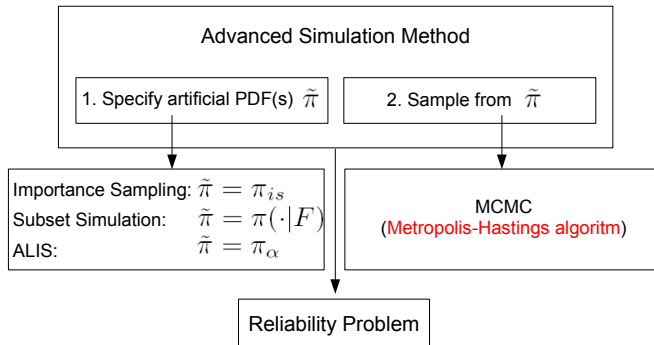
- N is very large, $N \sim 1000$
- p_F is very small, $p_F \sim 10^{-3} - 10^{-6}$
- π is the standard multidimensional Gaussian, $\pi = \mathcal{N}(0, I)$

We need to use Advanced Simulation Methods

Introduction

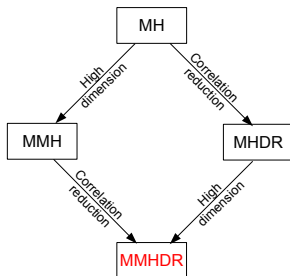


Introduction

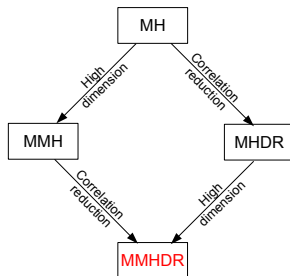


Main goal: to develop a novel effective MCMC algorithm for sampling from complex high-dimensional distributions

Modifications of the Metropolis-Hastings algorithm

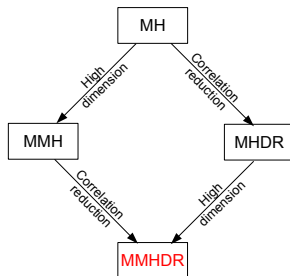


Modifications of the Metropolis-Hastings algorithm



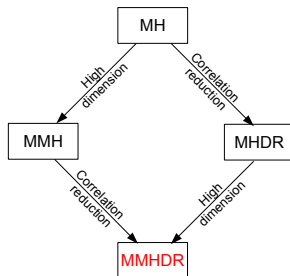
- Metropolis-Hastings ([MH](#)) algorithm, Metropolis et al. 1953, Hastings 1970

Modifications of the Metropolis-Hastings algorithm



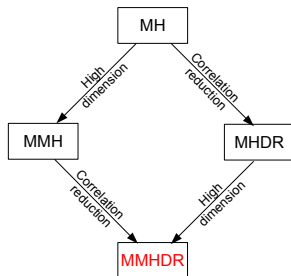
- Metropolis-Hastings ([MH](#)) algorithm, Metropolis et al. 1953, Hastings 1970
- Modified MH ([MMH](#)) algorithm, Au & Beck 2001
 - ▶ Au & Beck 2001: MH does not work in high dimensions
 - ▶ Katafygiotis & Zuev 2008: geometric explanation

Modifications of the Metropolis-Hastings algorithm



- Metropolis-Hastings ([MH](#)) algorithm, Metropolis et al. 1953, Hastings 1970
- Modified MH ([MMH](#)) algorithm, Au & Beck 2001
 - ▶ Au & Beck 2001: MH does not work in high dimensions
 - ▶ Katafygiotis & Zuev 2008: geometric explanation
- MH algorithm with delayed rejection ([MHDR](#)), Tierney & Mira 1999

Modifications of the Metropolis-Hastings algorithm



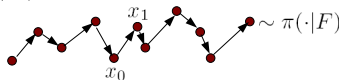
- Metropolis-Hastings (**MH**) algorithm, Metropolis et al. 1953, Hastings 1970
- Modified MH (**MMH**) algorithm, Au & Beck 2001
 - ▶ Au & Beck 2001: MH does not work in high dimensions
 - ▶ Katafygiotis & Zuev 2008: geometric explanation
- MH algorithm with delayed rejection (**MHDR**), Tierney & Mira 1999
- Modified Metropolis-Hastings algorithm with delayed rejection (**MMHDR**)

Standard Metropolis-Hastings algorithm

How to sample from $\pi(\cdot|F)$?

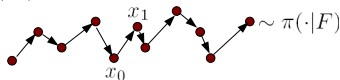
Standard Metropolis-Hastings algorithm

How to sample from $\pi(\cdot|F)$?



Standard Metropolis-Hastings algorithm

How to sample from $\pi(\cdot|F)$?

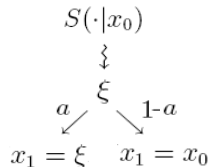


- 1 Simulate $\xi \sim S(\cdot|x_0)$
- 2 Compute the acceptance probability

$$a(x_0, \xi) = \min \left\{ 1, \frac{\pi(\xi)S(x_0|\xi)}{\pi(x_0)S(\xi|x_0)} I_F(\xi) \right\}$$

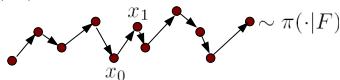
- 3 Accept/Reject ξ

$$x_1 = \begin{cases} \xi, & \text{with prob. } a(x_0, \xi); \\ x_0, & \text{with prob. } 1 - a(x_0, \xi). \end{cases}$$



Standard Metropolis-Hastings algorithm

How to sample from $\pi(\cdot|F)$?

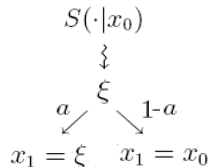


- 1 Simulate $\xi \sim S(\cdot|x_0)$
- 2 Compute the acceptance probability

$$a(x_0, \xi) = \min \left\{ 1, \frac{\pi(\xi)S(x_0|\xi)}{\pi(x_0)S(\xi|x_0)} I_F(\xi) \right\}$$

- 3 Accept/Reject ξ

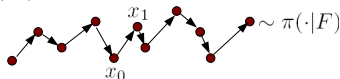
$$x_1 = \begin{cases} \xi, & \text{with prob. } a(x_0, \xi); \\ x_0, & \text{with prob. } 1 - a(x_0, \xi). \end{cases}$$



In high dimensions: $a(x_0, \xi) \approx 0 \Rightarrow x_1 = x_0$

Standard Metropolis-Hastings algorithm

How to sample from $\pi(\cdot|F)$?

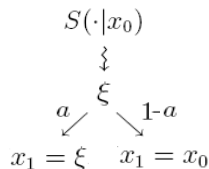


- 1 Simulate $\xi \sim S(\cdot|x_0)$
- 2 Compute the acceptance probability

$$a(x_0, \xi) = \min \left\{ 1, \frac{\pi(\xi)S(x_0|\xi)}{\pi(x_0)S(\xi|x_0)} I_F(\xi) \right\}$$

- 3 Accept/Reject ξ

$$x_1 = \begin{cases} \xi, & \text{with prob. } a(x_0, \xi); \\ x_0, & \text{with prob. } 1 - a(x_0, \xi). \end{cases}$$



In high dimensions: $a(x_0, \xi) \approx 0 \Rightarrow x_1 = x_0$

Reason: $S(\cdot|x_0)$ is N -dimensional PDF

Modified Metropolis-Hastings algorithm

1 Generate candidate state ξ

For each $j = 1 \dots N$:

- 1 Simulate $\hat{\xi}^j \sim S^j(\cdot|x_0^j)$
- 2 Compute the acceptance probability

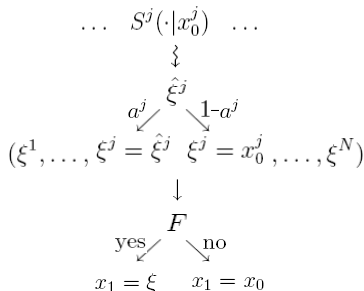
$$a^j(x_0^j, \hat{\xi}^j) = \min \left\{ 1, \frac{\pi_j(\hat{\xi}^j) S^j(x_0^j | \hat{\xi}^j)}{\pi_j(x_0^j) S^j(\hat{\xi}^j | x_0^j)} \right\}$$

- 3 Accept/Reject $\hat{\xi}^j$

$$\xi^j = \begin{cases} \hat{\xi}^j, & \text{with prob. } a^j(x_0^j, \hat{\xi}^j); \\ x_0^j, & \text{with prob. } 1 - a^j(x_0^j, \hat{\xi}^j). \end{cases}$$

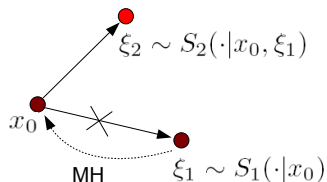
2 Accept/Reject ξ

$$x_1 = \begin{cases} \xi, & \text{if } \xi \in F; \\ x_0, & \text{if } \xi \notin F. \end{cases}$$

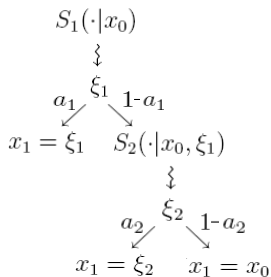
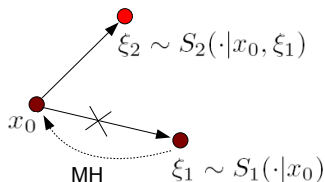


Metropolis-Hastings algorithm with Delayed Rejection

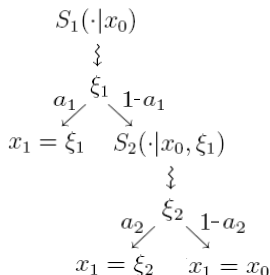
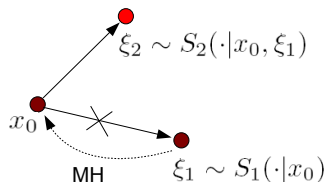
Metropolis-Hastings algorithm with Delayed Rejection



Metropolis-Hastings algorithm with Delayed Rejection



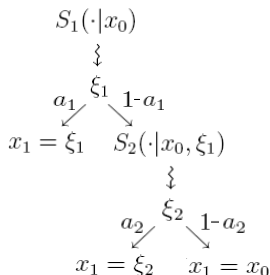
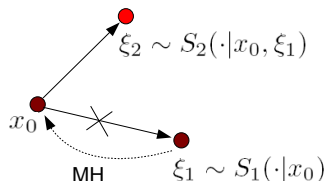
Metropolis-Hastings algorithm with Delayed Rejection



$$a_1(x_0, \xi_1) = \min \left\{ 1, \frac{\pi(\xi_1)S_1(x_0|\xi_1)}{\pi(x_0)S_1(\xi_1|x_0)} I_F(\xi_1) \right\}$$

$$a_2(x_0, \xi_1, \xi_2) = \min \left\{ 1, \frac{\pi(\xi_2)S_1(\xi_1|\xi_2)S_2(x_0|\xi_2, \xi_1)(1 - a_1(\xi_2, \xi_1))}{\pi(x_0)S_1(\xi_1|x_0)S_2(\xi_2|x_0, \xi_1)(1 - a_1(x_0, \xi_1))} I_F(\xi_2) \right\}$$

Metropolis-Hastings algorithm with Delayed Rejection

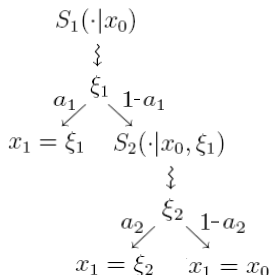
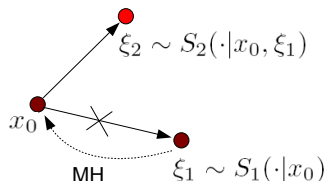


$$a_1(x_0, \xi_1) = \min \left\{ 1, \frac{\pi(\xi_1)S_1(x_0|\xi_1)}{\pi(x_0)S_1(\xi_1|x_0)} I_F(\xi_1) \right\}$$

$$a_2(x_0, \xi_1, \xi_2) = \min \left\{ 1, \frac{\pi(\xi_2)S_1(\xi_1|\xi_2)S_2(x_0|\xi_2, \xi_1)(1 - a_1(\xi_2, \xi_1))}{\pi(x_0)S_1(\xi_1|x_0)S_2(\xi_2|x_0, \xi_1)(1 - a_1(x_0, \xi_1))} I_F(\xi_2) \right\}$$

In high dimensions: $a_1(x_0, \xi_1) \approx a_2(x_0, \xi_1, \xi_2) \approx 0 \Rightarrow x_1 = x_0$

Metropolis-Hastings algorithm with Delayed Rejection



$$a_1(x_0, \xi_1) = \min \left\{ 1, \frac{\pi(\xi_1)S_1(x_0|\xi_1)}{\pi(x_0)S_1(\xi_1|x_0)} I_F(\xi_1) \right\}$$

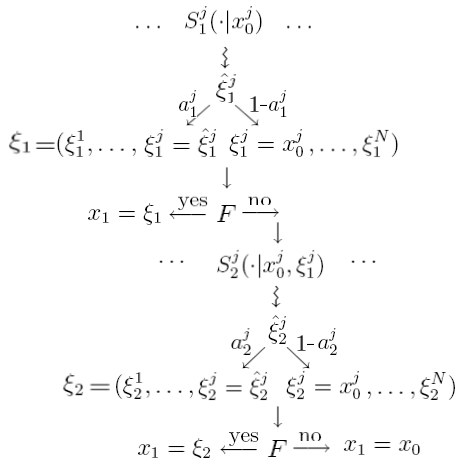
$$a_2(x_0, \xi_1, \xi_2) = \min \left\{ 1, \frac{\pi(\xi_2)S_1(\xi_1|\xi_2)S_2(x_0|\xi_2, \xi_1)(1 - a_1(\xi_2, \xi_1))}{\pi(x_0)S_1(\xi_1|x_0)S_2(\xi_2|x_0, \xi_1)(1 - a_1(x_0, \xi_1))} I_F(\xi_2) \right\}$$

In high dimensions: $a_1(x_0, \xi_1) \approx a_2(x_0, \xi_1, \xi_2) \approx 0 \Rightarrow x_1 = x_0$

Reason: $S_1(\cdot|x_0)$ and $S_2(\cdot|x_0, \xi_1)$ are N -dimensional PDFs

Modified MH algorithm with Delayed Rejection

$$x_0 \rightarrow x_1$$



Modified MH algorithm with Delayed Rejection

Features of the Algorithm

Modified MH algorithm with Delayed Rejection

Features of the Algorithm

- $S_2^j = S_2^j(\cdot | x_0^j, \xi_1^j)$

Modified MH algorithm with Delayed Rejection

Features of the Algorithm

- $S_2^j = S_2^j(\cdot | x_0^j, \xi_1^j)$
- Samples generated by MMHDR are **less correlated** than samples generated by MMH.

Modified MH algorithm with Delayed Rejection

Features of the Algorithm

- $S_2^j = S_2^j(\cdot | x_0^j, \xi_1^j)$
- Samples generated by MMHDR are **less correlated** than samples generated by MMH.
- MMHDR needs **more computational effort** than MMH for generating the same number of samples.

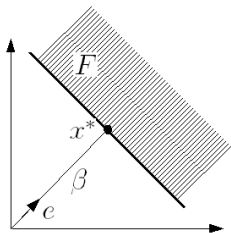
Modified MH algorithm with Delayed Rejection

Features of the Algorithm

- $S_2^j = S_2^j(\cdot | x_0^j, \xi_1^j)$
- Samples generated by MMHDR are **less correlated** than samples generated by MMH.
- MMHDR needs **more computational effort** than MMH for generating the same number of samples.
- With fixed computational effort:
 - ▶ MMH: more Markov chains with more correlated states
 - ▶ MMHDR: fewer Markov chains with less correlated states

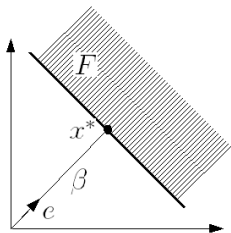
Example

Linear problem



Example

Linear problem

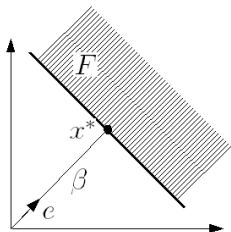


- Geometry

- ▶ $N = 1000$
- ▶ $p_F = 10^{-5}$, $\beta = 4.265$
- ▶ $x^* = e\beta$, $e \sim U(\mathbb{S}^{N-1})$

Example

Linear problem



- Geometry

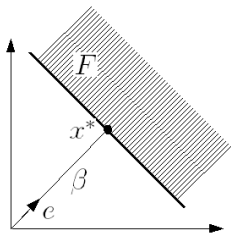
- ▶ $N = 1000$
- ▶ $p_F = 10^{-5}$, $\beta = 4.265$
- ▶ $x^* = e\beta$, $e \sim U(\mathbb{S}^{N-1})$

- Proposal PDFs

- ▶ MMH: $S^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$
- ▶ MMHDR: $S_{1,2}^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$

Example

Linear problem



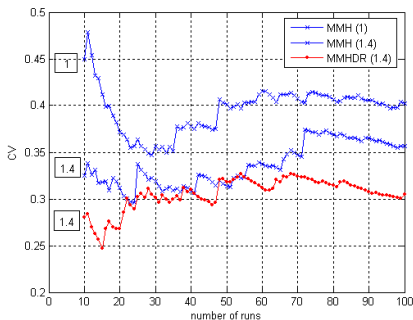
- Geometry

- ▶ $N = 1000$
- ▶ $p_F = 10^{-5}$, $\beta = 4.265$
- ▶ $x^* = e\beta$, $e \sim U(\mathbb{S}^{N-1})$

- Proposal PDFs

- ▶ MMH: $S^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$
- ▶ MMHDR: $S_{1,2}^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$

Subset Simulation



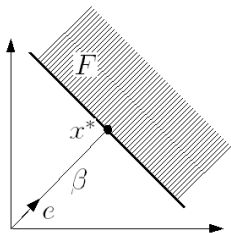
MMH(1): SS + MMH, $n = 10^3$

MMHDR(1.4): SS + MMHDR, $n = 10^3$

MMH(1.4): SS + MMH, $n = 1450$

Example

Linear problem



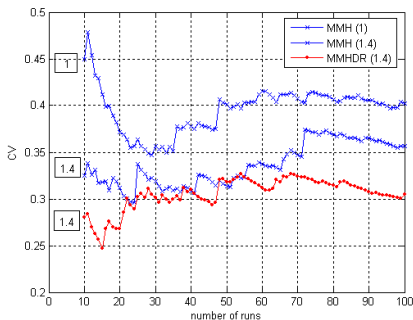
- Geometry

- ▶ $N = 1000$
- ▶ $p_F = 10^{-5}$, $\beta = 4.265$
- ▶ $x^* = e\beta$, $e \sim U(\mathbb{S}^{N-1})$

- Proposal PDFs

- ▶ MMH: $S^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$
- ▶ MMHDR: $S_{1,2}^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$

Subset Simulation



MMH(1): SS + MMH, $n = 10^3$

MMHDR(1.4): SS + MMHDR, $n = 10^3$

MMH(1.4): SS + MMH, $n = 1450$

Reduction in CV is 11%

Summary

Summary

- We propose a novel modification of the Metropolis-Hastings algorithm, called **Modified Metropolis-Hastings algorithm with Delayed Rejection**

Summary

- We propose a novel modification of the Metropolis-Hastings algorithm, called **M**odified **M**etropolis-**H**astings algorithm with **D**elayed **R**ejection
- **MMHDR** = **MMH** (Au & Beck 2001) + **MHDR** (Tierney & Mira 1999)

Summary

- We propose a novel modification of the Metropolis-Hastings algorithm, called **Modified Metropolis-Hastings algorithm with Delayed Rejection**
- $\text{MMHDR} = \text{MMH}$ (Au & Beck 2001) + MHDR (Tierney & Mira 1999)
- MMHDR is designed for sampling from **high dimensional conditional** PDFs

Summary

- We propose a novel modification of the Metropolis-Hastings algorithm, called **Modified Metropolis-Hastings algorithm with Delayed Rejection**
- **MMHDR** = **MMH** (Au & Beck 2001) + **MHDR** (Tierney & Mira 1999)
- **MMHDR** is designed for sampling from **high dimensional conditional** PDFs
- The efficiency of the **MMHDR** is demonstrated with a numerical example

Thank You for attention!

