Markov chain Monte Carlo and its Application to some Engineering Problems

Konstantin Zuev

Department of Computing & Mathematical Sciences
California Institute of Technology

http://www.its.caltech.edu/∼zuev

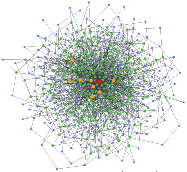
May 24, 2011

Mechanical & Civil Engineering Seminar, Caltech

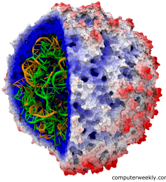
MCMC Revolution

P. Diaconis (2009), "The Markov chain Monte Carlo revolution":

> *...asking about applications of Markov chain Monte Carlo (MCMC) is a little like asking about applications of the quadratic formula... you can take any area of science, from hard to social, and find a burgeoning MCMC literature specifically tailored to that area.*



slowmuse.wordpress.com



computerweekly.com

The main goal of this talk: To demonstrate how MCMC algorithms can be used for solving engineering problems

Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
3. How does MCMC work?
4. MCMC applications to Reliability Problem
5. Summary

Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
3. How does MCMC work?
4. MCMC applications to Reliability Problem
5. Summary

Purpose of MCMC

Problem: To estimate

$$I = \int_{\Theta} h(\theta)\pi(\theta)d\theta$$

- $\Theta \subseteq \mathbb{R}^d$ parameter space
- $h : \Theta \to \mathbb{R}$ function of interest
- $\pi(\theta)$ "target" PDF on Θ

Purpose of MCMC

Problem: To estimate

$$I = \int_{\Theta} h(\theta)\pi(\theta)d\theta = \mathbb{E}_{\pi}[h]$$

- $\Theta \subseteq \mathbb{R}^d$ parameter space
- $h : \Theta \to \mathbb{R}$ function of interest
- $\pi(\theta)$ "target" PDF on Θ

Purpose of MCMC

Problem: To estimate

$$I = \int_\Theta h(\theta)\pi(\theta)d\theta = \mathbb{E}_\pi[h]$$

- $\Theta \subseteq \mathbb{R}^d$ parameter space
- $h : \Theta \to \mathbb{R}$ function of interest
- $\pi(\theta)$ "target" PDF on Θ

"Easy" Cases:

- d is small $(d = 1, 2, 3) \Rightarrow$ numerical integration
- $\pi(\theta)$ is easy to sample from $\Rightarrow$ Monte Carlo method

Purpose of MCMC

Problem: To estimate

$$I = \int_{\Theta} h(\theta)\pi(\theta)d\theta = \mathbb{E}_{\pi}[h]$$

- $\Theta \subseteq \mathbb{R}^d$ parameter space
- $h : \Theta \to \mathbb{R}$ function of interest
- $\pi(\theta)$ "target" PDF on Θ

"Easy" Cases:

- d is small $(d = 1, 2, 3)$ $\Rightarrow$ numerical integration
- $\pi(\theta)$ is easy to sample from $\Rightarrow$ Monte Carlo method

Typical Case in Applications:

- d is large $(d \sim 10^2 - 10^3)$
- $\pi(\theta)$ is known only up to a constant, $\pi(\theta) = cf(\theta)$

Purpose of MCMC

Problem: To estimate

$$I = \int_{\Theta} h(\theta)\pi(\theta)d\theta = \mathbb{E}_{\pi}[h]$$

- $\Theta \subseteq \mathbb{R}^d$ parameter space
- $h : \Theta \to \mathbb{R}$ function of interest
- $\pi(\theta)$ "target" PDF on Θ

"Easy" Cases:

- d is small $(d = 1, 2, 3) \Rightarrow$ numerical integration
- $\pi(\theta)$ is easy to sample from $\Rightarrow$ Monte Carlo method

Typical Case in Applications:

- d is large $(d \sim 10^2 - 10^3)$
- $\pi(\theta)$ is known only up to a constant, $\pi(\theta) = cf(\theta)$

Solution: Use an appropriate MCMC method

Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
   - Bayesian Inference
   - Optimal Stochastic Design
   - Reliability Problem
3. How does MCMC work?
4. MCMC applications to Reliability Problem
5. Summary

Bayesian Inference

- $\mathcal{M}$ the assumed model class for the target dynamic system:
  - set of I/O probability models $p(y|\theta, u)$ $\xrightarrow{u} \boxed{\text{System}} \xrightarrow{y}$
  - $\theta \in \Theta$ the uncertain model parameters
  - prior PDF $\pi_0(\theta)$ over Θ

Bayesian Inference

- $\mathcal{M}$ the assumed model class for the target dynamic system:
  - set of I/O probability models $p(y|\theta, u)$ $\xrightarrow{u} \boxed{\text{System}} \xrightarrow{y}$
  - $\theta \in \Theta$ the uncertain model parameters
  - prior PDF $\pi_0(\theta)$ over Θ

Bayesian approach:

- Update $\pi_0(\theta)$ to posterior PDF $\pi(\theta|\mathcal{D})$ via Bayes' theorem:

$$\pi(\theta|\mathcal{D}) = L(\mathcal{D}|\theta)\pi_0(\theta)/\mathcal{Z}$$

- $\mathcal{D}$ the measured data from the system
- $L(\mathcal{D}|\theta)$ the likelihood function

Bayesian Inference

- $\mathcal{M}$ the assumed model class for the target dynamic system:
  - set of I/O probability models $p(y|\theta, u)$ $\xrightarrow{u}$ $\boxed{\text{System}}$ $\xrightarrow{y}$
  - $\theta \in \Theta$ the uncertain model parameters
  - prior PDF $\pi_0(\theta)$ over Θ

Bayesian approach:

- Update $\pi_0(\theta)$ to posterior PDF $\pi(\theta|\mathcal{D})$ via Bayes' theorem:

$$\pi(\theta|\mathcal{D}) = L(\mathcal{D}|\theta)\pi_0(\theta)/\mathcal{Z}$$

- $\mathcal{D}$ the measured data from the system
- $L(\mathcal{D}|\theta)$ the likelihood function

Problems:

- Evidence

$$\mathcal{Z} = \int_\Theta L(\mathcal{D}|\theta)\pi_0(\theta)d\theta$$

- Posterior expectations

$$\mathbb{E}_\pi[h] = \int_\Theta h(\theta)\pi(\theta|\mathcal{D})d\theta$$

Optimal Stochastic Design Problem

- $\mathcal{M}$ the assumed model class for the target dynamic system
- $\theta \in \Theta$ the uncertain model parameters
- $\varphi \in \Phi$ design variables: controllable parameters that define the system design

Optimal Stochastic Design Problem

- $\mathcal{M}$ the assumed model class for the target dynamic system
- $\theta \in \Theta$ the uncertain model parameters
- $\varphi \in \Phi$ design variables: controllable parameters that define the system design
- The objective function for a robust-to-uncertainties design:

$$\mathbb{E}_\pi[h(\varphi,\theta)] = \int_\Theta h(\varphi,\theta)\pi(\theta|\varphi)d\theta$$

- $h(\varphi,\theta)$ a performance function of the system, e.g. stress in a component
- $\pi(\theta|\varphi)$ a PDF, which incorporates available knowledge about the system

Optimal Stochastic Design Problem

- $\mathcal{M}$ the assumed model class for the target dynamic system
- $\theta \in \Theta$ the uncertain model parameters
- $\varphi \in \Phi$ design variables: controllable parameters that define the system design
- The objective function for a robust-to-uncertainties design:

$$\mathbb{E}_\pi[h(\varphi, \theta)] = \int_\Theta h(\varphi, \theta)\pi(\theta|\varphi)d\theta$$

- $h(\varphi, \theta)$ a performance function of the system, e.g. stress in a component
- $\pi(\theta|\varphi)$ a PDF, which incorporates available knowledge about the system

Problem:

$$\varphi^* = \arg\min_{\varphi \in \Phi} \mathbb{E}_\pi[h(\varphi, \theta)]$$

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ represents the uncertain input load
  - θ is a stochastic vector and has joint PDF π
- $F \subset \mathbb{R}^d$ a failure domain (unacceptable performance)

$$F = \{\theta : G(\theta) \geq b^*\}$$

- $G(\theta)$ a performance function
- b^* a critical threshold for performance
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
3. How does MCMC work?
   - Markov chains
   - Markov chain Monte Carlo
   - Metropolis-Hastings algorithm
4. MCMC applications to Reliability Problem
5. Summary

MCMC: The Main Idea

- Monte Carlo method

$$\int_{\Theta} h(\theta)\pi(\theta)d\theta \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_i), \quad \theta_i \overset{i.i.d}{\sim} \pi$$

MCMC: The Main Idea

- Monte Carlo method

$$\int_{\Theta} h(\theta)\pi(\theta)d\theta \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_i), \quad \theta_i \overset{i.i.d}{\sim} \pi$$

- How to obtain i.i.d. samples from π ?

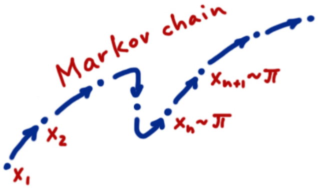
MCMC: The Main Idea

- Monte Carlo method

$$\int_{\Theta} h(\theta)\pi(\theta)d\theta \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_i), \quad \theta_i \overset{i.i.d}{\sim} \pi$$

- How to obtain i.i.d. samples from π ?

- MCMC samples from π and computes integrals using Markov chains:



$$\int_{\Theta} h(\theta)\pi(\theta)d\theta \approx \frac{1}{N-N_0}\sum_{i=N_0+1}^{N} h(x_i)$$

Markov chains

Markov chains

- $K(x, \mathbb{R}^d) \equiv 1$
- $K(x, \{x\})$ is not necessarily zero

Markov chains

- $K(x, \mathbb{R}^d) \equiv 1$
- $K(x, \{x\})$ is not necessarily zero

Definition

Markov chain is a sequence of stochastic vectors $x_0, x_1, x_2, \dots$ such that

$$P(x_{n+1} \in A | x_0, \dots, x_n) = P(x_{n+1} \in A | x_n) \equiv K(x_n, A)$$

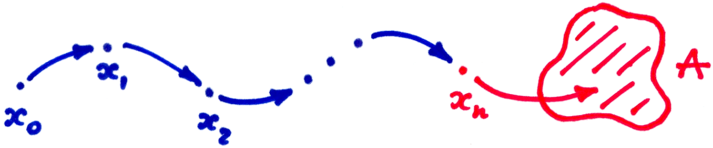
Markov chains

- $K(x, \mathbb{R}^d) \equiv 1$
- $K(x, \{x\})$ is not necessarily zero

Definition

Markov chain is a sequence of stochastic vectors $x_0, x_1, x_2, \ldots$ such that

$$P(x_{n+1} \in A | x_0, \ldots, x_n) = P(x_{n+1} \in A | x_n) \equiv K(x_n, A)$$



Markov chains

Markov chain theory: To describe the distribution of x_n when $n \to \infty$

Markov chains

Markov chain theory: To describe the distribution of x_n when $n \to \infty$

Definition

A probability distribution π is called stationary distribution for K if

$$\pi(A) = \int K(x, A)\pi(x)dx, \quad \text{for all measurable sets } A$$

Markov chains

Markov chain theory: To describe the distribution of x_n when $n \to \infty$

Definition

A probability distribution π is called stationary distribution for K if

$$\pi(A) = \int K(x, A)\pi(x)dx, \quad \text{for all measurable sets } A$$

The central result: Let K be a transition kernel with a stationary distribution π where K satisfies certain "ergodic conditions", then

- π is the unique stationary distribution
- $x_n \sim \pi$, when $n \to \infty$

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

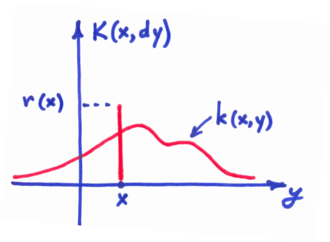
The main goal: **To find an appropriate transition kernel K**

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: **To find an appropriate transition kernel K**

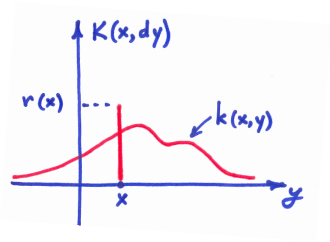


Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: **To find an appropriate transition kernel K**



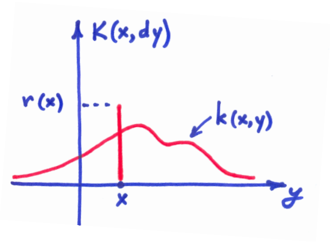
- $K(x,dy) = k(x,y)dy + r(x)\delta_x(dy)$

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: To find an appropriate transition kernel K



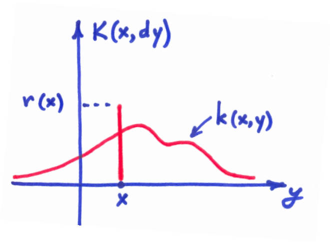
- $K(x, dy) = k(x, y)dy + r(x)\delta_x(dy)$
- Jumps from x to $y \neq x$ occur according to $k(x, y)$

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: **To find an appropriate transition kernel K**



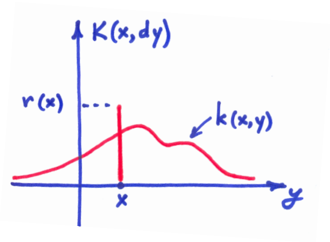
- $K(x,dy) = k(x,y)dy + r(x)\delta_x(dy)$
- Jumps from x to $y \neq x$ occur according to $k(x,y)$
- $r(x) = 1 - \int k(x,y)dy$
  probability that the chain remains at x

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: **To find an appropriate transition kernel K**



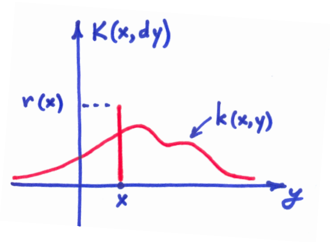
- $K(x, dy) = k(x, y)dy + r(x)\delta_x(dy)$
- Jumps from x to $y \neq x$ occur according to $k(x, y)$
- $r(x) = 1 - \int k(x, y)dy$
  probability that the chain remains at x
- $\delta_x(dy) = 1$ if $x \in dy$ and 0 otherwise

Markov chain Monte Carlo

MCMC methods: The stationary distribution π is given:

it is the target distribution we want to sample from

The main goal: **To find an appropriate transition kernel K**



- $K(x, dy) = k(x, y)dy + r(x)\delta_x(dy)$
- Jumps from x to $y \neq x$ occur according to $k(x, y)$
- $r(x) = 1 - \int k(x, y)dy$ probability that the chain remains at x
- $\delta_x(dy) = 1$ if $x \in dy$ and 0 otherwise

The central result: If k satisfies the detailed balance equation

$$\pi(x)k(x, y) = \pi(y)k(y, x)$$

then π is the stationary distribution for K

Metropolis-Hastings algorithm

$$\boxed{k_{MH}(x,y) = S(y|x)a(x,y)}$$

- $S(y|x)$ proposal distribution
- $a(x,y)$ acceptance probability

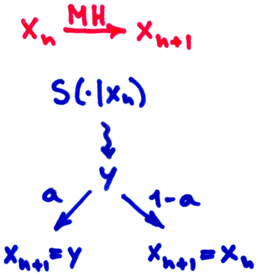
$$a(x,y) = \min\left\{1, \frac{\pi(y)S(x|y)}{\pi(x)S(y|x)}\right\}$$

Metropolis-Hastings algorithm

$$\boxed{k_{MH}(x,y) = S(y|x)a(x,y)}$$

- $S(y|x)$ proposal distribution
- $a(x,y)$ acceptance probability

$$a(x,y) = \min\left\{1, \frac{\pi(y)S(x|y)}{\pi(x)S(y|x)}\right\}$$

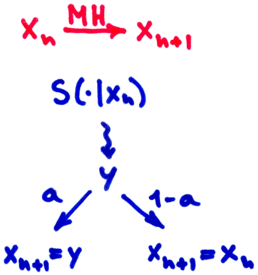


Metropolis-Hastings algorithm

$$\boxed{k_{MH}(x,y) = S(y|x)a(x,y)}$$

- $S(y|x)$ proposal distribution
- $a(x,y)$ acceptance probability

$$a(x,y) = \min\left\{1, \frac{\pi(y)S(x|y)}{\pi(x)S(y|x)}\right\}$$



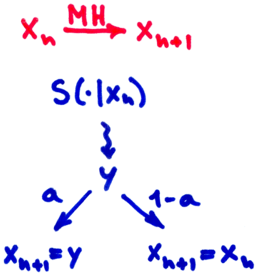
- N. Metropolis et al, 1953
  - Statistical mechanics
  - Boltzmann distribution
  - $S(x|y) = S(y|x)$
  - Absence of Markov chains

Metropolis-Hastings algorithm

$$\boxed{k_{MH}(x,y) = S(y|x)a(x,y)}$$

- $S(y|x)$ proposal distribution
- $a(x,y)$ acceptance probability

$$a(x,y) = \min\left\{1, \frac{\pi(y)S(x|y)}{\pi(x)S(y|x)}\right\}$$



- N. Metropolis et al, 1953
  - Statistical mechanics
  - Boltzmann distribution
  - $S(x|y) = S(y|x)$
  - Absence of Markov chains
- W.K. Hastings, 1970
  - Generalization of the M-algorithm
  - Markov chains come into play
  - Non-symmetric proposals
  - Different acceptance probabilities

Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
3. How does MCMC work?
4. MCMC applications to Reliability Problem
    - Subset Simulation
    - Enhancements for Subset Simulation
5. Summary

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int\limits_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000$

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000 \Rightarrow$ numerical quadrature methods are not suitable

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000 \Rightarrow$ numerical quadrature methods are not suitable
- We can calculate $I_F(\theta)$ for any θ, but it is expensive (dyn. system analysis)

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000 \Rightarrow$ numerical quadrature methods are not suitable
- We can calculate $I_F(\theta)$ for any θ, but it is expensive (dyn. system analysis)
- p_F is very small, $p_F \sim 10^{-3} - 10^{-6}$

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int\limits_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000 \Rightarrow$ numerical quadrature methods are not suitable
- We can calculate $I_F(\theta)$ for any θ, but it is expensive (dyn. system analysis)
- p_F is very small, $p_F \sim 10^{-3} - 10^{-6} \Rightarrow$ Monte Carlo method is not suitable

Reliability Problem

Reliability Problem: To estimate the probability of failure p_F

$$p_F = P(\theta \in F) = \int_{\mathbb{R}^d} \pi(\theta) I_F(\theta) d\theta$$

Notation:

- $\theta \in \mathbb{R}^d$ is a stochastic vector with the joint PDF π
- $F \subset \mathbb{R}^d$ is a failure domain, $F = \{\theta : G(\theta) \geq b^*\}$
- $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$

Typically in Applications:

- d is very large, $d \sim 1000 \Rightarrow$ numerical quadrature methods are not suitable
- We can calculate $I_F(\theta)$ for any θ, but it is expensive (dyn. system analysis)
- p_F is very small, $p_F \sim 10^{-3} - 10^{-6} \Rightarrow$ Monte Carlo method is not suitable

We need MCMC based simulation methods

Subset Simulation

S.K. Au & J.L. Beck (PEM, 2001)

Subset Simulation

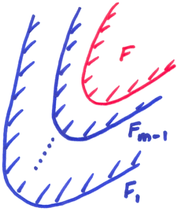
S.K. Au & J.L. Beck (PEM, 2001)

$$\mathbb{R}^d = F_0 \supset F_1 \supset \ldots \supset F_m = F$$

$$F = \{\theta : G(\theta) \geq b^*\}$$

$$F_i = \{\theta : G(\theta) \geq b_i^*\}$$

$$b_1^* < b_2^* < \ldots < b_m^* = b^*$$



$$p_F = \prod_{k=0}^{m-1} P(F_{k+1}|F_k)$$

Subset Simulation

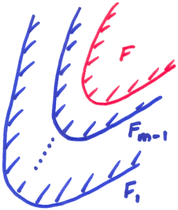
S.K. Au & J.L. Beck (PEM, 2001)

$$\mathbb{R}^d = F_0 \supset F_1 \supset \ldots \supset F_m = F$$

$$F = \{\theta : G(\theta) \geq b^*\}$$

$$F_i = \{\theta : G(\theta) \geq b_i^*\}$$

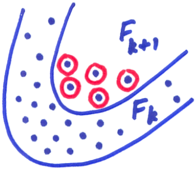
$$b_1^* < b_2^* < \ldots < b_m^* = b^*$$



$$p_F = \prod_{k=0}^{m-1} P(F_{k+1}|F_k)$$

$$P(F_{k+1}|F_k) \approx \frac{1}{N}\sum_{i=1}^{N} I_{F_{k+1}}(\theta_k^{(i)})$$

$$\theta_k^{(i)} \sim \pi(\theta|F_k) = \frac{\pi(\theta) I_{F_k}(\theta)}{P(F_k)}$$



Subset Simulation

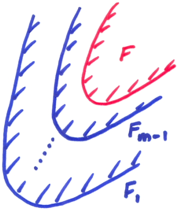
S.K. Au & J.L. Beck (PEM, 2001)

$$\mathbb{R}^d = F_0 \supset F_1 \supset \ldots \supset F_m = F$$

$$F = \{\theta : G(\theta) \geq b^*\}$$

$$F_i = \{\theta : G(\theta) \geq b_i^*\}$$

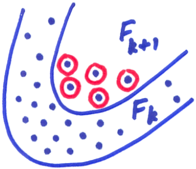
$$b_1^* < b_2^* < \ldots < b_m^* = b^*$$



$$p_F = \prod_{k=0}^{m-1} P(F_{k+1}|F_k)$$

$$P(F_{k+1}|F_k) \approx \frac{1}{N}\sum_{i=1}^{N} I_{F_{k+1}}(\theta_k^{(i)})$$

$$\theta_k^{(i)} \sim \pi(\theta|F_k) = \frac{\pi(\theta) I_{F_k}(\theta)}{P(F_k)}$$



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

Subset Simulation

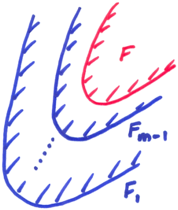
S.K. Au & J.L. Beck (PEM, 2001)

$$\mathbb{R}^d = F_0 \supset F_1 \supset \ldots \supset F_m = F$$

$$F = \{\theta : G(\theta) \geq b^*\}$$

$$F_i = \{\theta : G(\theta) \geq b_i^*\}$$

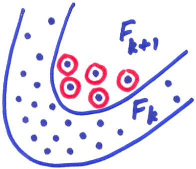
$$b_1^* < b_2^* < \ldots < b_m^* = b^*$$



$$p_F = \prod_{k=0}^{m-1} P(F_{k+1}|F_k)$$

$$P(F_{k+1}|F_k) \approx \frac{1}{N}\sum_{i=1}^{N} I_{F_{k+1}}(\theta_k^{(i)})$$

$$\theta_k^{(i)} \sim \pi(\theta|F_k) = \frac{\pi(\theta)I_{F_k}(\theta)}{P(F_k)}$$



- How to sample from $\pi(\cdot|F_k)$?
- Use an appropriate MCMC alg

Modified Metropolis-Hastings algorithm

S.K. Au & J.L. Beck (PEM, 2001): Standard Metropolis-Hastings algorithm is not efficient in high dimensions

Modified Metropolis-Hastings algorithm

S.K. Au & J.L. Beck (PEM, 2001):

Standard Metropolis-Hastings algorithm is not efficient in high dimensions

- Generate candidate state y

  For each $j = 1 \dots N$:

  - Simulate $\hat{y}^j \sim S^j(\cdot|x_n^j)$
  - Compute the acceptance probability

$$a^j(x_n^j, \hat{y}^j) = \min\left\{1, \frac{\pi_j(\hat{y}^j)}{\pi_j(x_n^j)}\right\}$$

  - Accept/Reject $\hat{y}^j$

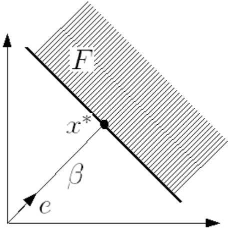
$$y^j = \begin{cases} \hat{y}^j, & \text{with prob. } a^j(x_n^j, \hat{y}^j) \\ x_n^j, & \text{with prob. } 1 - a^j(x_n^j, \hat{y}^j) \end{cases}$$

- Accept/Reject y

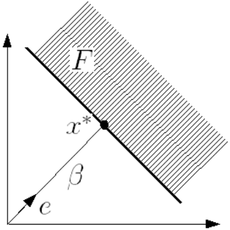
$$x_{n+1} = \begin{cases} y, & \text{if } y \in F_i \\ x_n, & \text{if } y \notin F_i \end{cases}$$

Efficiency of Subset Simulation

- Linear Problem
  - $F \subset \mathbb{R}^d$ is a hyperplane
  - $d = 1000$
  - $p_F = 10^{-k}$, $k = 3, 4, 5, 6$



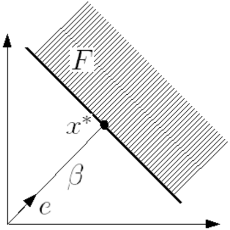
Efficiency of Subset Simulation



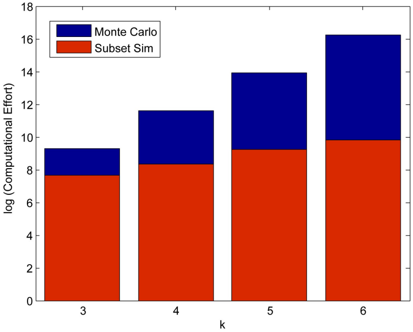
- Linear Problem
  - $F \subset \mathbb{R}^d$ is a hyperplane
  - $d = 1000$
  - $p_F = 10^{-k}$, $k = 3, 4, 5, 6$
- What computational effort is required to achieve the COV $\delta = 30\%$?

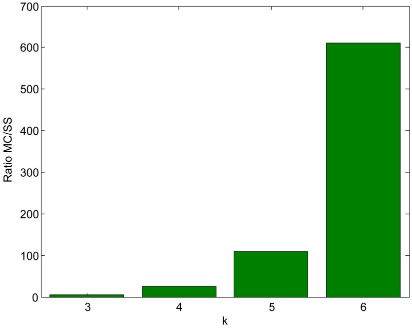
Efficiency of Subset Simulation

- Linear Problem
  - $F \subset \mathbb{R}^d$ is a hyperplane
  - $d = 1000$
  - $p_F = 10^{-k}$, $k = 3, 4, 5, 6$



- What computational effort is required to achieve the COV $\delta = 30\%$?

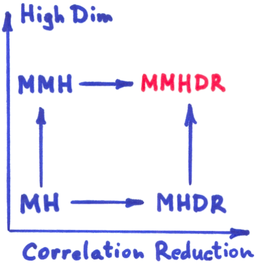




Outline

1. What problems is MCMC meant to solve?
2. Why is MCMC useful in Engineering?
3. How does MCMC work?
4. MCMC applications to Reliability Problem
    - Subset Simulation
    - Enhancements for Subset Simulation
        - Modified Metropolis-Hastings algorithm with Delayed Rejection
        - Bayesian post-processor for Subset Simulation
5. Summary

Modifications of the Metropolis-Hastings algorithm



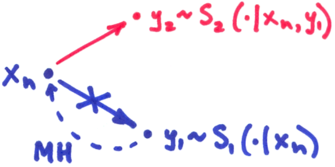
- Modified MH (MMH) algorithm
  - S.K. Au & J.L. Beck, 2001
- MH algorithm with delayed rejection (MHDR)
  - L. Tierney & A. Mira, 1999
- Modified Metropolis-Hastings algorithm with delayed rejection (MMHDR)
  - K.M. Zuev & L.S. Katafygiotis, 2011

Metropolis-Hastings algorithm with Delayed Rejection

L. Tierney & A. Mira (Statistics in Medicine, 1999):

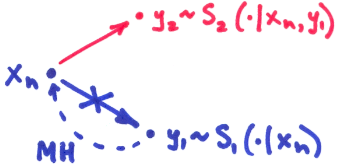
Metropolis-Hastings algorithm with Delayed Rejection

L. Tierney & A. Mira (Statistics in Medicine, 1999):



Metropolis-Hastings algorithm with Delayed Rejection

L. Tierney & A. Mira (Statistics in Medicine, 1999):

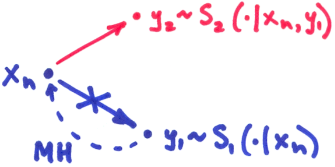


$$a_1(x_n, y_1) = \min\left\{1, \frac{\pi(y_1)}{\pi(x_n)} I_F(y_1)\right\}$$

$$a_2(x_n, y_1, y_2) = \min\left\{1, \frac{\pi(y_2) S_1(y_1|y_2)(1 - a_1(y_2, y_1))}{\pi(x_n) S_1(y_1|x_n)(1 - a_1(x_n, y_1))} I_F(y_2)\right\}$$

Metropolis-Hastings algorithm with Delayed Rejection

L. Tierney & A. Mira (Statistics in Medicine, 1999):



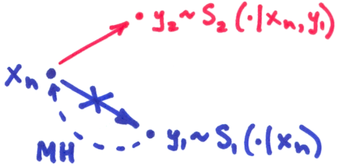
$$a_1(x_n, y_1) = \min\left\{1, \frac{\pi(y_1)}{\pi(x_n)} I_F(y_1)\right\}$$

$$a_2(x_n, y_1, y_2) = \min\left\{1, \frac{\pi(y_2)S_1(y_1|y_2)(1 - a_1(y_2, y_1))}{\pi(x_n)S_1(y_1|x_n)(1 - a_1(x_n, y_1))} I_F(y_2)\right\}$$

- Drawback: Inefficient in high dimensions

Metropolis-Hastings algorithm with Delayed Rejection

L. Tierney & A. Mira (Statistics in Medicine, 1999):

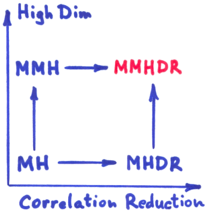


$$a_1(x_n, y_1) = \min\left\{1, \frac{\pi(y_1)}{\pi(x_n)} I_F(y_1)\right\}$$

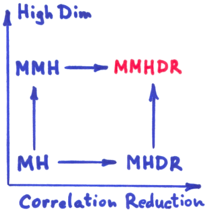
$$a_2(x_n, y_1, y_2) = \min\left\{1, \frac{\pi(y_2) S_1(y_1|y_2)(1 - a_1(y_2, y_1))}{\pi(x_n) S_1(y_1|x_n)(1 - a_1(x_n, y_1))} I_F(y_2)\right\}$$

- Drawback: Inefficient in high dimensions
- Reason: $S_1(\cdot|x_n)$ and $S_2(\cdot|x_n, y_1)$ are d-dimensional PDFs

Modified MH algorithm with Delayed Rejection

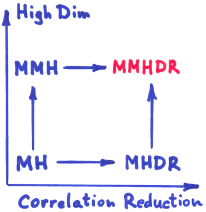


Modified MH algorithm with Delayed Rejection



Features of the Algorithm:

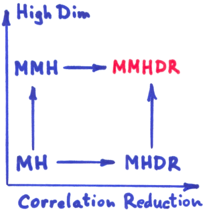
Modified MH algorithm with Delayed Rejection



Features of the Algorithm:

- Samples generated by MMHDR are less correlated then samples generated by MMH.

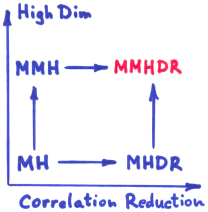
Modified MH algorithm with Delayed Rejection



Features of the Algorithm:

- Samples generated by MMHDR are less correlated then samples generated by MMH.
- MMHDR needs more computational effort than MMH for generating the same number of samples.

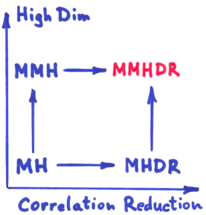
Modified MH algorithm with Delayed Rejection



Features of the Algorithm:

- Samples generated by MMHDR are less correlated then samples generated by MMH.
- MMHDR needs more computational effort than MMH for generating the same number of samples.
- Whether MMHDR is useful for reliability problems depends on whether the gained reduction in variance compensates for the additional cost.

Modified MH algorithm with Delayed Rejection

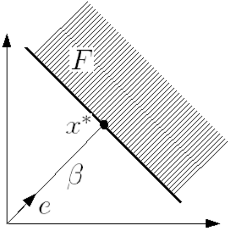


Features of the Algorithm:

- Samples generated by MMHDR are less correlated then samples generated by MMH.
- MMHDR needs more computational effort than MMH for generating the same number of samples.
- Whether MMHDR is useful for reliability problems depends on whether the gained reduction in variance compensates for the additional cost.
- With fixed computational effort:
  - MMH: more Markov chains with more correlated states
  - MMHDR: fewer Markov chains with less correlated states

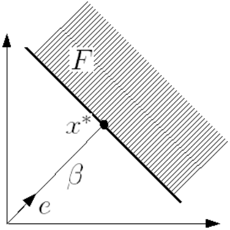
Example

- Linear problem



Example

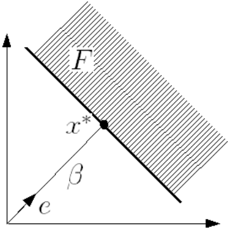
- Linear problem



- Geometry
  - $d = 1000$
  - $p_F = 10^{-5}$, $\beta = 4.265$

Example

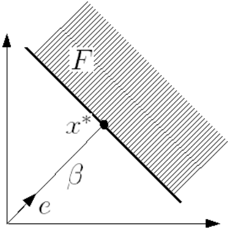
- Linear problem



- Geometry
  - $d = 1000$
  - $p_F = 10^{-5}$, $\beta = 4.265$
- Proposal PDFs
  - MMH: $S^j(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$
  - MMHDR: $S^j_{1,2}(\cdot|x_0^j) = \mathcal{N}(x_0^j, 1)$

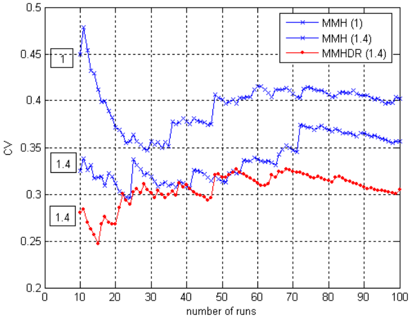
Example

- Linear problem



- Geometry
  - $d = 1000$
  - $p_F = 10^{-5}$, $\beta = 4.265$
- 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



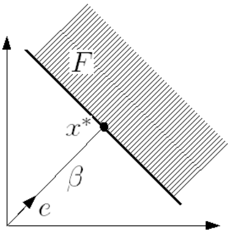
MMH(1): SS + MMH, $n = 10^3$

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

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

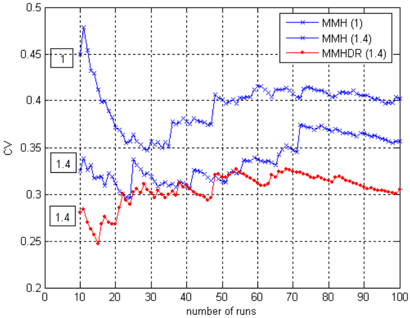
Example

- Linear problem



- Geometry
  - $d = 1000$
  - $p_F = 10^{-5}$, $\beta = 4.265$
- 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%

“Bayesianization” of Subset Simulation

"Bayesianization" of Subset Simulation

The key idea of SS:

$$p_F = \prod_{k=1}^{m} p_k, \qquad p_k = P(F_k | F_{k-1})$$

"Bayesianization" of Subset Simulation

The key idea of SS:

$$p_F = \prod_{k=1}^{m} p_k, \qquad p_k = P(F_k | F_{k-1})$$

Original ("frequentist") SS:

$$p_k \approx \hat{p}_k = \frac{1}{N} \sum_{i=1}^{N} I_{F_k}(\theta_{k-1}^{(i)}) = \frac{n_k}{N}, \qquad p_F \approx \hat{p}_F^{SS} = \prod_{k=1}^{m} \frac{n_k}{N}$$

"Bayesianization" of Subset Simulation

The key idea of SS:

$$p_F = \prod_{k=1}^{m} p_k, \qquad p_k = P(F_k|F_{k-1})$$

Original ("frequentist") SS:

$$p_k \approx \hat{p}_k = \frac{1}{N}\sum_{i=1}^{N} I_{F_k}(\theta_{k-1}^{(i)}) = \frac{n_k}{N}, \qquad p_F \approx \hat{p}_F^{SS} = \prod_{k=1}^{m} \frac{n_k}{N}$$

Bayesian SS:

1. Specify prior PDFs $p(p_k)$ for all $p_k = P(F_k|F_{k-1})$, $k = 1, \ldots, m$.

"Bayesianization" of Subset Simulation

The key idea of SS:

$$p_F = \prod_{k=1}^{m} p_k, \qquad p_k = P(F_k|F_{k-1})$$

Original ("frequentist") SS:

$$p_k \approx \hat{p}_k = \frac{1}{N}\sum_{i=1}^{N} I_{F_k}(\theta_{k-1}^{(i)}) = \frac{n_k}{N}, \qquad p_F \approx \hat{p}_F^{SS} = \prod_{k=1}^{m} \frac{n_k}{N}$$

Bayesian SS:

1. Specify prior PDFs $p(p_k)$ for all $p_k = P(F_k|F_{k-1})$, $k = 1, \ldots, m$.
2. Find the posterior PDFs $p(p_k|\mathcal{D}_{k-1})$ via Bayes' theorem, using new data $\mathcal{D}_{k-1} = \{\theta_{k-1}^{(1)}, \ldots, \theta_{k-1}^{(N)} \sim \pi(\cdot|F_{k-1})\}$

"Bayesianization" of Subset Simulation

The key idea of SS:

$$p_F = \prod_{k=1}^{m} p_k, \qquad p_k = P(F_k|F_{k-1})$$

Original ("frequentist") SS:

$$p_k \approx \hat{p}_k = \frac{1}{N}\sum_{i=1}^{N} I_{F_k}(\theta_{k-1}^{(i)}) = \frac{n_k}{N}, \qquad p_F \approx \hat{p}_F^{SS} = \prod_{k=1}^{m} \frac{n_k}{N}$$

Bayesian SS:

1. Specify prior PDFs $p(p_k)$ for all $p_k = P(F_k|F_{k-1})$, $k = 1, \ldots, m$.
2. Find the posterior PDFs $p(p_k|\mathcal{D}_{k-1})$ via Bayes' theorem, using new data $\mathcal{D}_{k-1} = \{\theta_{k-1}^{(1)}, \ldots, \theta_{k-1}^{(N)} \sim \pi(\cdot|F_{k-1})\}$
3. Obtain the posterior PDF $p(p_F| \cup_{k=0}^{m-1} \mathcal{D}_k)$ of $p_F = \prod_{k=1}^{m} p_k$ from $p(p_1|\mathcal{D}_0), \ldots, p(p_m|\mathcal{D}_{m-1})$.

Prior and Posterior for $p_k = P(F_k|F_{k-1})$

Prior and Posterior for $p_k = P(F_k|F_{k-1})$

1. Prior PDF $p(p_k)$

   Principle of Maximum Entropy:

$$p(p_k) = 1, \qquad 0 \leq p_k \leq 1.$$

Prior and Posterior for $p_k = P(F_k|F_{k-1})$

1. Prior PDF $p(p_k)$

   Principle of Maximum Entropy:

$$p(p_k) = 1, \qquad 0 \leq p_k \leq 1.$$

2. Posterior PDF $p(p_k|\mathcal{D}_{k-1})$

Prior and Posterior for $p_k = P(F_k|F_{k-1})$

1. Prior PDF $p(p_k)$

   Principle of Maximum Entropy:

$$p(p_k) = 1, \qquad 0 \le p_k \le 1.$$

2. Posterior PDF $p(p_k|\mathcal{D}_{k-1})$

   - If $\theta_{k-1}^{(1)}, \dots, \theta_{k-1}^{(N)}$ are i.i.d. according to $\pi(\cdot|F_{k-1})$
     $\Rightarrow I_{F_k}(\theta_{k-1}^{(1)}), \dots, I_{F_k}(\theta_{k-1}^{(N)})$ can be interpreted as Bernoulli trials
     $\Rightarrow$ Bayes' Theorem (1763):

$$p(p_k|\mathcal{D}_{k-1}) = \frac{p_k^{n_k}(1-p_k)^{N-n_k}}{B(n_k+1, N-n_k+1)}$$

Prior and Posterior for $p_k = P(F_k|F_{k-1})$

1. Prior PDF $p(p_k)$

   Principle of Maximum Entropy:

$$p(p_k) = 1, \qquad 0 \leq p_k \leq 1.$$

2. Posterior PDF $p(p_k|\mathcal{D}_{k-1})$

   - If $\theta_{k-1}^{(1)}, \ldots, \theta_{k-1}^{(N)}$ are i.i.d. according to $\pi(\cdot|F_{k-1})$
     $\Rightarrow I_{F_k}(\theta_{k-1}^{(1)}), \ldots, I_{F_k}(\theta_{k-1}^{(N)})$ can be interpreted as Bernoulli trials
     $\Rightarrow$ Bayes' Theorem (1763):

$$p(p_k|\mathcal{D}_{k-1}) = \frac{p_k^{n_k}(1-p_k)^{N-n_k}}{B(n_k+1, N-n_k+1)}$$

   - In fact, $\theta_{k-1}^{(1)}, \ldots, \theta_{k-1}^{(N)}$ are MCMC samples (for $k \geq 2$)
     $\Rightarrow \theta_{k-1}^{(1)}, \ldots, \theta_{k-1}^{(N)} \sim \pi(\cdot|F_{k-1})$, however, they are not independent

$$p(p_k|\mathcal{D}_{k-1}) \approx \frac{p_k^{n_k}(1-p_k)^{N-n_k}}{B(n_k+1, N-n_k+1)}$$

Posterior PDF for p_F

Last step: To find the PDF of $p_F = \prod_{k=1}^{m} p_k$, given the PDFs of all factors

$$p_k \sim \mathcal{B}e(n_k + 1, N - n_k + 1)$$

Posterior PDF for p_F

Last step: To find the PDF of $p_F = \prod_{k=1}^{m} p_k$, given the PDFs of all factors

$$p_k \sim \mathcal{B}e(n_k + 1, N - n_k + 1)$$

Idea: To approximate p_F by a single beta variable

Posterior PDF for p_F

Last step: To find the PDF of $p_F = \prod_{k=1}^{m} p_k$, given the PDFs of all factors

$$p_k \sim \mathcal{B}e(n_k + 1, N - n_k + 1)$$

Idea: To approximate p_F by a single beta variable

Theorem (Da-Yin Fan, 1991)

Let $X_1, \ldots, X_m$ be beta variables, $X_k \sim Beta(a_k, b_k)$, and $Y = X_1 X_2 \ldots X_m$. Then Y is approximately distributed as $\tilde{Y} \sim Beta(a, b)$, where

$$a = \mu_1 \frac{\mu_1 - \mu_2}{\mu_2 - \mu_1^2}, \qquad b = (1 - \mu_1)\frac{\mu_1 - \mu_2}{\mu_2 - \mu_1^2},$$

$$\mu_1 = \prod_{k=1}^{m} \frac{a_k}{a_k + b_k}, \qquad \mu_2 = \prod_{k=1}^{m} \frac{a_k(a_k + 1)}{(a_k + b_k)(a_k + b_k + 1)}.$$

Posterior PDF for p_F

Last step: To find the PDF of $p_F = \prod_{k=1}^{m} p_k$, given the PDFs of all factors

$$p_k \sim \mathcal{B}e(n_k + 1, N - n_k + 1)$$

Idea: To approximate p_F by a single beta variable

Theorem (Da-Yin Fan, 1991)

Let $X_1, \ldots, X_m$ be beta variables, $X_k \sim Beta(a_k, b_k)$, and $Y = X_1 X_2 \ldots X_m$. Then Y is approximately distributed as $\tilde{Y} \sim Beta(a, b)$, where

$$a = \mu_1 \frac{\mu_1 - \mu_2}{\mu_2 - \mu_1^2}, \qquad b = (1 - \mu_1)\frac{\mu_1 - \mu_2}{\mu_2 - \mu_1^2},$$

$$\mu_1 = \prod_{k=1}^{m} \frac{a_k}{a_k + b_k}, \qquad \mu_2 = \prod_{k=1}^{m} \frac{a_k(a_k + 1)}{(a_k + b_k)(a_k + b_k + 1)}.$$

Nice property of this approximation: $\mathbb{E}[\tilde{Y}] = \mathbb{E}[Y]$, $\mathbb{E}[\tilde{Y}^2] = \mathbb{E}[Y^2]$

Bayesian post-processor for Subset Simulation

$$\boxed{\text{Point estimate } \hat{p}_F^{SS} \rightsquigarrow \text{ PDF } p^{SS+}(p_F) = \mathcal{B}e(p_F|a,b)}$$

Bayesian post-processor for Subset Simulation

$$\boxed{\text{Point estimate } \hat{p}_F^{SS} \rightsquigarrow \text{ PDF } p^{SS+}(p_F) = \mathcal{B}e(p_F|a,b)}$$

$$a = \frac{\prod_{k=1}^{m} \frac{n_k+1}{N+2}\left(1 - \prod_{k=1}^{m} \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m} \frac{n_k+2}{N+3} - \prod_{k=1}^{m} \frac{n_k+1}{N+2}} \quad b = \frac{\left(1 - \prod_{k=1}^{m} \frac{n_k+1}{N+2}\right)\left(1 - \prod_{k=1}^{m} \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m} \frac{n_k+2}{N+3} - \prod_{k=1}^{m} \frac{n_k+1}{N+2}}$$

Bayesian post-processor for Subset Simulation

$$\boxed{\text{Point estimate } \hat{p}_F^{SS} \rightsquigarrow \text{ PDF } p^{SS+}(p_F) = \mathcal{B}e(p_F|a,b)}$$

$$a = \frac{\prod_{k=1}^m \frac{n_k+1}{N+2}\left(1 - \prod_{k=1}^m \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^m \frac{n_k+2}{N+3} - \prod_{k=1}^m \frac{n_k+1}{N+2}} \quad b = \frac{\left(1 - \prod_{k=1}^m \frac{n_k+1}{N+2}\right)\left(1 - \prod_{k=1}^m \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^m \frac{n_k+2}{N+3} - \prod_{k=1}^m \frac{n_k+1}{N+2}}$$

- What is the relationship between p^{SS+} and $\hat{p}_F^{SS}$?

Bayesian post-processor for Subset Simulation

$$\boxed{\text{Point estimate } \hat{p}_F^{SS} \rightsquigarrow \text{ PDF } p^{SS+}(p_F) = \mathcal{B}e(p_F|a,b)}$$

$$a = \frac{\prod_{k=1}^{m}\frac{n_k+1}{N+2}\left(1-\prod_{k=1}^{m}\frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m}\frac{n_k+2}{N+3}-\prod_{k=1}^{m}\frac{n_k+1}{N+2}} \quad b = \frac{\left(1-\prod_{k=1}^{m}\frac{n_k+1}{N+2}\right)\left(1-\prod_{k=1}^{m}\frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m}\frac{n_k+2}{N+3}-\prod_{k=1}^{m}\frac{n_k+1}{N+2}}$$

- What is the relationship between p^{SS+} and $\hat{p}_F^{SS}$?

$$\mathbb{E}_{p^{SS+}}[p_F] \to \hat{p}_F^{SS}, \text{ as } N \to \infty$$

$$\mathbb{E}_{p^{SS+}}[p_F] \approx \hat{p}_F^{SS}, \text{ when } N \text{ is large}$$

Bayesian post-processor for Subset Simulation

$$\boxed{\text{Point estimate } \hat{p}_F^{SS} \rightsquigarrow \text{ PDF } p^{SS+}(p_F) = \mathcal{B}e(p_F|a,b)}$$

$$a = \frac{\prod_{k=1}^{m} \frac{n_k+1}{N+2}\left(1 - \prod_{k=1}^{m} \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m} \frac{n_k+2}{N+3} - \prod_{k=1}^{m} \frac{n_k+1}{N+2}} \quad b = \frac{\left(1 - \prod_{k=1}^{m} \frac{n_k+1}{N+2}\right)\left(1 - \prod_{k=1}^{m} \frac{n_k+2}{N+3}\right)}{\prod_{k=1}^{m} \frac{n_k+2}{N+3} - \prod_{k=1}^{m} \frac{n_k+1}{N+2}}$$

- What is the relationship between p^{SS+} and $\hat{p}_F^{SS}$?

$$\mathbb{E}_{p^{SS+}}[p_F] \rightarrow \hat{p}_F^{SS}, \text{ as } N \rightarrow \infty$$

$$\mathbb{E}_{p^{SS+}}[p_F] \approx \hat{p}_F^{SS}, \text{ when } N \text{ is large}$$

- The PDF p^{SS+} can be fully used for life-cost analyses, decision making, etc.

$$\mathbb{E}[\mathrm{Loss}(p_F)] = \int \mathrm{Loss}(p_F) p^{SS+}(p_F) dp_F$$

Elasto-Plastic Structure Subjected to Ground Motion

S.K. Au (Computers & Structures, 2005):

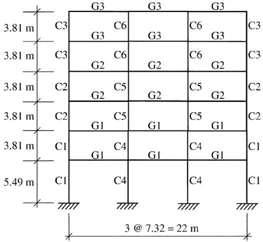
- 2D moment-resisting steel frame
- Synthetic ground motion $a = a(Z)$
  - $Z = (Z_1, \ldots, Z_d) \overset{i.i.d}{\sim} \mathcal{N}(0,1)$
  - $\xrightarrow{Z} \boxed{\text{Filter}} \xrightarrow{a(Z)}$
  - $d = 1001$
- Failure domain:

$$F = \{Z \in \mathbb{R}^d : \delta_{\max}(Z) > b\}$$

$$\delta_{\max} = \max_{i=1,\ldots,6} \delta_i$$

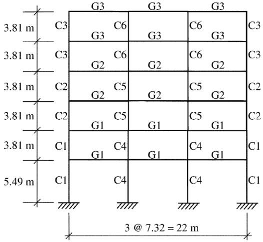
δ_i is the maximum absolute interstory drift ratio of the i^{th} story within the duration of study, 30 s

$$b = 0.5\% \Rightarrow p_F \approx 8.9 \times 10^{-3}$$



Elasto-Plastic Structure Subjected to Ground Motion

S.K. Au (Computers & Structures, 2005):



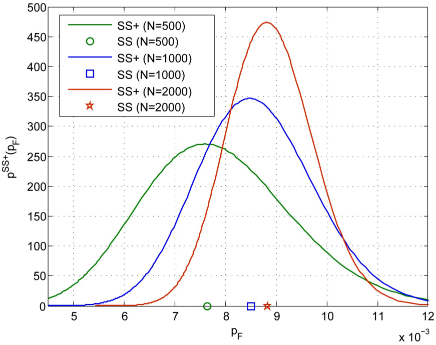
- 2D moment-resisting steel frame
- Synthetic ground motion $a = a(Z)$
  - $Z = (Z_1, \dots, Z_d) \overset{i.i.d}{\sim} \mathcal{N}(0,1)$
  - $\xrightarrow{Z}$ $\boxed{\text{Filter}}$ $\xrightarrow{a(Z)}$
  - $d = 1001$
- Failure domain:

$$F = \{Z \in \mathbb{R}^d : \delta_{\max}(Z) > b\}$$

$$\delta_{\max} = \max_{i=1,\dots,6} \delta_i$$

δ_i is the maximum absolute interstory drift ratio of the i^{th} story within the duration of study, 30 s

$$b = 0.5\% \Rightarrow p_F \approx 8.9 \times 10^{-3}$$



Summary

- MCMC algorithms computes multi-dimensional integrals using Markov chains

Summary

- MCMC algorithms computes multi-dimensional integrals using Markov chains
- MCMC-based methods can be very useful for solving engineering problems
  - Reliability Engineering

Summary

- MCMC algorithms computes multi-dimensional integrals using Markov chains
- MCMC-based methods can be very useful for solving engineering problems
  - Reliability Engineering
- Subset Simulation (Au & Beck, 2001)
  - a very efficient MCMC method for estimation of small failure probabilities

Summary

- MCMC algorithms computes multi-dimensional integrals using Markov chains
- MCMC-based methods can be very useful for solving engineering problems
  - Reliability Engineering
- Subset Simulation (Au & Beck, 2001)
  - a very efficient MCMC method for estimation of small failure probabilities
- Enhancements for Subset Simulation
  - MMHDR = MMH (Au & Beck, 2001) + MHDR (Tierney & Mira, 1999)
  - Bayesian post-processor for Subset Simulation

Acknowledgements

- Jim Beck (Caltech)
  - for teaching me Bayesian way of thinking
  - for mentoring my research at Caltech

- Lambros Katafygiotis (HKUST)
  - for introducing me to the world of MCMC algorithms
  - for supervising me at HKUST

- Ivan Au (CityU)
  - for providing Matlab code for the nonlinear example
  - for being always open for constructive discussions

Thank you for attention!