

# Fundamental Concepts in Computational and Applied Mathematics

Juan Meza  
School of Natural Sciences  
University of California, Merced

Fall 2014

# Short History of Monte Carlo

- Original Metropolis algorithm dates back to 1953 paper (J. Chem Phys) for integrals in  $\mathbb{R}^{2N}$ , where  $N$  denoted the number of particles in a problem of interest
- Hastings proposed a solution to the "curse of dimensionality" in 1970 paper[1] .
- Gelfand and Smith [2] (1990) brought more attention to the subject
- Advent of more computational power, especially parallel processing makes MC methods more popular
- Combination of computing power and algorithms drives new interest in Bayesian statistics

# Motivating applications

Example 1: Generate samples from a given probability distribution function (pdf),  $f(x)$

$$X_1, X_2, \dots, X_t$$

Example 2: Computation of certain integrals:

$$I = \int f(x) \, dx$$

Example 3: Given  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

find  $x_* \in \mathbb{R}^n$  such that  $f(x_*) \leq f(x), \forall x \in \mathbb{R}^n$

# Some important assumptions & facts

- We will usually assume that we can compute the pdf  $f(x)$  only up to a multiplicative constant
- Don't always know the normalizing constant
- Sampling from a given distribution can be quite difficult, especially in the high-dimensional case
- We should pick more samples from regions where  $f(x)$  is "big", but how do we know that without knowing where  $f$  is "big"

# Acceptance/Rejection Method

Given  $f(x) \leq Mg(x)$

**repeat**

    Generate  $X \sim g$  and  $u \sim U(0, 1)$

**if**  $u \leq f(X)/Mg(X)$  **then**

        |  $Y = X$

**end**

**until** (*done*);

**Algorithm 1:** Acceptance-Rejection (AR)

## Remark

Algorithm (AR) produces a variable  $Y$  distributed according to  $f(x)$ .

# Integration Example

Suppose we want to compute the following integral (expectation):

$$E[f(X)] = \frac{\int f(x)\pi(x) \, dx}{\int \pi(x) \, dx},$$

and the distribution of  $X$  is in  $k$ -dimensional Euclidian space.

First we compute samples  $(X_t, t = 1, \dots, N)$  with the given distribution  $f(x)$  and then approximate

$$E[f(X)] = \frac{1}{n} \sum_{t=1}^n f(X_t).$$

## Remark

Accuracy will depend on distribution, number of samples we generate, and dimension of the problem. The last is known as the "curse of dimensionality"

# Metropolis-Hastings (1970)

Let

$$\alpha(X, Y) = \min \left( 1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)} \right).$$

Initialize  $X_0$ ; set  $t = 0$

**while** (*not done*) **do**

    Generate  $Y \sim q(\cdot|X_t); u \sim U(0, 1)$

**if**  $u \leq \alpha(X_t, Y)$  **then**

$X_{t+1} = Y$

**else**

$X_{t+1} = X_t$

**end**

    Increment  $t$

**end**

**Algorithm 2:** Metropolis-Hastings (MH)

## Terminology

$q(\cdot|X)$  is called the proposal distribution.

# Ising Model Part 1

Suppose we have a  $2D$  array of spins  $\sigma_i \in [-1, 1]$  and an energy function defined for a configuration of a system

$$E(\sigma) = - \sum_{ij} \mathcal{J}_{ij} \sigma_i \sigma_j - B \sum_k \sigma_k$$

and we want to take the mean of some function  $f(\sigma)$ :

$$\mathcal{F} = \frac{1}{Z(T)} \sum_{\sigma} f(\sigma) \exp(-E(\sigma)/\kappa T),$$

where  $Z(T) = \sum_{\sigma} \exp(-E(\sigma)/\kappa T)$  is the *partition function*



## Ising Model Part 2

Ideally would like to select configurations from the distribution

$$g(\sigma) = \frac{\exp(-E(\sigma)/\kappa T)}{Z(T)},$$

and we could approximate the mean by:

$$F = \frac{\sum_k f(\sigma_k)}{M}$$

### Remark

Original Metropolis algorithm observed that if only one spin is changed, sampling  $g$  is easy. Showed that if a move lowers the energy accept it. If it raises the energy accept it with probability  $p$ . Use AR algorithm to choose site for move.

# Simulated Annealing Method

```
Initialize  $X_0$ ; set  $t = 0$   
while (not converged) do  
  Generate new state  $Y$  and  $u \sim U(0, 1)$   
  if  $E(Y) < E(X_t)$  then  
    |  $X_{t+1} = Y$   
  else  
    | if  $u < e^{-[(E(Y)-E(X_n))/T]}$  then  
      |  $X_{t+1} = Y$   
    | end  
  end  
end
```

## Algorithm 3: Simulated Annealing (AR)

### Remark

Simulated Annealing always takes a point where function is lower. But it will also accept a higher function value with a certain probability depending on the "temperature"  $T$ .

# Basic Markov Chain Facts

- A Markov Chain is a sequence of random variables
- Constructed from a transition kernel  $K$ , which is a conditional probability density s.t.  $X_{n+1} \sim K(X_n, X_{n+1})$
- Has stability property in that a *stationary probability distribution* exists by construction, i.e. if  $X_n \sim \pi$  then  $X_{n+1} \sim \pi$
- Homogeneous is when transition probability is independent of state – hence if the initial distribution or initial state is known, then the construction of MC is entirely dependent on transition

# Basic Idea for Markov Chain Monte Carlo

- Use properties of Markov Chains to construct a transition kernel that has a stationary probability distribution that matches the one we want
- Assume stationarity, irreducibility, and aperiodic, we can prove Ergodic Theorem.
- Needs to also satisfy reversibility condition  $\pi_i p(i \rightarrow j) = \pi_j p(j \rightarrow i)$

# Practicalities

- Need to have a "burn-in" period
- How long do you run the Markov Chain
- Can use multiple chains to accelerate convergence

# Summary

- Metropolis algorithm defines a Markov Chain whose limit is the desired probability distribution
- Many variations on the proposal distribution
- Growing number of applications

# References I



W.K. Hastings,

*Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Biometrika, Vol. 57, No. 1., pp. 97-109, 1970



A.E. Gelfand and A.F.M. Smith,

*Sampling-Based Approaches to Calculating Marginal Densities*,  
Journal of the American Statistical Association, Vol. 85, No. 410.  
(Jun., 1990), pp. 398-409



I. Beichi, F. Sullivan

The Metropolis Algorithm, Computing in Science and Engineering,  
January/February 2000.