Introduction to Markov Chain Monte Carlo Techniques

Rajeeva Karandikar

rkarandikar@gmail.com

rlk@cmi.ac.in

corrected version of the version used for the talk, includes 6 additional pages.

As I mentioned in the abstract, there are several examples where statistical or probabilistic ideas have played a role in proving a result in mathematics which has nothing to do with probability theory. Sometimes, the alternate proof via probabilistic ideas may be simpler.

Weierstrass Theorem on approximation of a continuous function on the unit interval by polynomials is one such:

Weierstrass proved in 1888 that every continuous function f on $[0,1]$ can be uniformly approximated by polynomials. In 1912, Bernstein gave an alternate proof. In his words, his proof used Calculus of Probabilities.

Probability theory - which was still evolving (This was about 20 years before Kolmogorov's book)!

Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités
(Demonstration of a theorem of Weierstrass based on the calculus of probabilities),
by Sergei Bernstein, Communications of the Kharkov Mathematical Society, Volume XIII,
1912/13, p. 1-2)}

Let us consider another such example: Consider a $N \times N$ chessboard. Each square is assigned a number 1 or 0. 1 means the square is occupied and 0 means that the square is unoccupied.

Each such assignment is called a configuration. A configuration is said to be feasible if all the neighbours of every occupied square are unoccupied. (Every square that is not in the first or last row or column has 8 neighbours.)

Thus a configuration is feasible if for every pair of adjacent squares, at most one square has a 1 .

For a feasible configuration (denoted by Γ),

let $f(\Gamma)$ denote the number of 1's in Γ.

The quantity of interest is the average of $f(\Gamma)$ where the average is taken over

the uniform distribution over the set of feasible configurations.

Denoting the feasible configurations by $\mathcal{S}$ the object of interest is

$$\alpha = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} f(\Gamma)$$

The total number of configurations is 2^{N*N} and even when $N = 25$, this number is 2^{625} so it is not computationally feasible to scan all configurations, count the feasible configurations and take the average of $f(\Gamma)$.

It is easy to see that when N=25, the total number of feasible configurations is at least 2^{169}. To see this, assign 0 to all squares whose one of the coordinates is even (the squares are indexed from 1 to 15). In the remaining 64 positions, we can assign a 1 or 0. It is clear that each such configuration is feasible and the total number of such configurations is 2^{169}. Clearly, this is only a lower bound, the total will be a lot more. Thus computationally, listing all feasible configurations is not possible!

Indeed, along with

$$\alpha = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} f(\Gamma)$$

another quantity of interest is weighted average of $f(\Gamma)$ with weights being proportional to $\exp\{-Kf(\Gamma)\}$: for some $0 < K < \infty$, namely,

$$\beta = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} c \exp\{-Kf(\Gamma)\} f(\Gamma)$$

where the constant c is such that $c \sum_{\Gamma \in \#\mathcal{S}} \exp\{-Kf(\Gamma)\} = 1$.

How does one compute (or approximate) α and β?

Markov Chain: Let S be a finite set and let $\{X_n : n \geq 0\}$ be a sequence of random variables such that for all $n \geq 0$, $i, j, i_0, i_1, ..., i_{n-1} \in S$, we have

$$P(X_{n+1} = j \mid X_0 = i_0, \ldots X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j \mid X_n = i) = p_{ij}$$

The matrix $P = ((p_{ij}))$ is called the transition probability matrix of the Markov Chain $\{X_n\}$. Denoting the n^{th} power of the matrix P by $P^n = ((p_{ij}^{(n)}))$, it can be seen that

$$P(X_n = j \mid X_0 = i) = P(X_{n+m} = j \mid X_m = i) = p_{ij}^{(n)}.$$

Assuming that $\{X_n : n \geq 0\}$ is irreducible and aperiodic

(i) $\forall i, j \in S$, $\exists n \geq 1$ such that $p_{ij}^{(n)} > 0$,

(ii) $\forall i \in S$, $g.c.d.\{n \geq 1 : p_{ii}^{(n)} > 0\} = 1$,

it follows that for all $i, j \in S$

$$\lim_n p_{ij}^{(n)} = \pi_j$$

where $\{\pi_j : j \in S\}$ is the unique eigenvector for the eigenvalue 1 of P such that $\sum_j \pi_j = 1$.

The result

$$\lim_n p_{ij}^{(n)} = \pi_j$$

where $\{\pi_j : j \in S\}$ is the unique eigenvector for the eigenvalue 1 of P such that $\sum_j \pi_j = 1$ is a Classical result on Markov Chains and has several interesting proofs. Also follows from Perron–Frobenius theorem on positive matrices.

How does one identify $\{\pi_j : j \in S\}$?

It can be expressed explicitly using Graph theory (the technique is called Markov Chain Tree Theorem, due to F.T. Leighton and R.L. Rivest 1983).

Of course $\sum_j p_{ij} = 1$ for all $i \in S$.

Simplest case : if

$$\sum_i p_{ij} = 1 \text{ for all } j \in S,$$

then it is easy to see that $\pi_j = \frac{1}{m}$ where $m = \#S$.

Such a P is called doubly stochastic.

The law of large numbers gives that for a irreducible and aperiodic Marov Chain $\{X_n : n \geq 0\}$:

for any function $f : S \mapsto \mathbb{R}$, we have

$$\lim_n \frac{1}{n} \sum_{t=1}^{n} f(X_t) = \sum_{j \in S} f(j)\pi_j.$$

It is a version of Ergodic theorem, also known as time average equals the space average.

Thus returning to our problem of $n \times n$ chess board, if we can get a irreducible aperiodic doubly stochastic transition probability matrix P on $\mathcal{S}$, then $\{\pi_\Gamma\}$ given by

$$\pi_\Gamma = \frac{1}{\#\mathcal{S}}$$

and hence

$$\lim_n \frac{1}{n} \sum_{t=1}^{n} f(X_t) = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} f(\Gamma).$$

Thus all we need to do in our first problem is to get a transition function $p_{\Gamma,\Lambda}$ on the class of feasible configurations such that it is irreducible aperiodic and doubly stochastic. Then simulating $\{X_t : t = 0, 1, 2, \ldots, n\}$ for large n we can assert that $\lim_n \frac{1}{n} \sum_{t=1}^{n} f(X_t)$ approximates α.

How can we get such a $p_{\Gamma,\Lambda}$ when we cannot even find the cardinality of the set of feasible configurations?

Let us note that while we do not know the cardinality of the set of feasible configurations, given a feasible configuration Λ, we can list the set of feasible configurations Γ that are adjacent to Λ.

Let us describe a transition function $p_{\Gamma,\Lambda}$ as follows:

Given a feasible configuration $\Gamma \in \mathcal{S}$, choose a square s (out of the N^2 squares) with equal probability. If any of the neighbors of s is occupied (has 1) then $\Lambda = \Gamma$; if all the neighbors of s are unoccupied (have 0) then flip the state of the square s.

While we can write an expression for $p_{\Gamma,\Lambda}$, it is very easy to write a code to implement the same.

It is easy to see that transition function $\{p_{\Gamma,\Lambda}\}$ is irreducible aperiodic and $p_{\Gamma,\Lambda} = p_{\Gamma,\Lambda}$ and hence it is doubly stochastic. Thus

$$\lim_n \frac{1}{n}\sum_{t=1}^{n} f(X_t) = \frac{1}{\#\mathcal{S}}\sum_{\Gamma\in\mathcal{S}} f(\Gamma).$$

So we can approximate $\alpha = \frac{1}{\#\mathcal{S}}\sum_{\Gamma\in\mathcal{S}} f(\Gamma)$ without knowing how large is $\mathcal{S}$!

Interesting question: How large should n be to ensure that

$$\left|\frac{1}{\#\mathcal{S}}\sum_{\Gamma\in\mathcal{S}} f(\Gamma) - \lim_{n}\frac{1}{n}\sum_{t=1}^{n} f(X_t)\right|$$

is small?

The answer depends upon the probability transition function.

The answer depends upon (among other factors) the smallest m such that $p_{ij}^{(m)} > 0$ for all $i, j \in \mathbb{S}$.

In our problem of $n \times n$ chess board, it can be seen that $m \leq 2n^2$ works *i.e.* $p_{\Gamma,\Lambda}^{(2n*n)} > 0$ for all feasible configurations Γ, Λ, while the number $\#S$ grows faster than $e^{\frac{m}{4}}$.

Thus $p_{\Gamma,\Lambda}$ is a good probability transition function for our problem of approximating α.

So we need to simulate $\{X_t : 1 \leq t \leq m\}$ for a very large m and then take

$$\hat{\alpha}_m = \lim_m \frac{1}{m} \sum_{t=1}^{m} f(X_t)$$

as the estimate of α.

This idea was used in the Manhattan project and was named Monte Caro technique in an article by Ulam and Metropolis in 1949.

In the chess board problem, we have already noted that it is easy to simulate X_{t+1} once we know what is X_t. So we need to choose X_0 and then start...

With current hardware, we can compute $\hat{\alpha}_m$ with m being million rather quickly.

$\hat{\alpha}_m$ does depend upon X_0. What do we do about it?

What can we do to approximate

$$\beta = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} c \exp\{-K f(\Gamma)\} f(\Gamma).$$

Can we get hold of a suitable transition probability matrix $q_{\Gamma,\Lambda}$ so that

$$\pi^*_\Gamma = c \exp\{-K f(\Gamma)\}$$

is the unique eigenvector of $q_{\Gamma,\Lambda}$ for eigenvalue 1?

Then it would follow that limit of n^{th} power of the transition probability matrix $q_{\Gamma,\Lambda}$ is π^*_Γ

If

$$\pi^*_{\Gamma} = c \exp\{-K f(\Gamma)\}$$

is the unique eigenvector of $q_{\Gamma,\Lambda}$ for eigenvalue 1, then it would follow that limit of n^{th} power of the transition probability matrix is π^*_{Γ} and if $\{X_t\}$ is the Markov chain with transition probabilities $q_{\Gamma,\Lambda}$, then

$$\lim_n \frac{1}{n} \sum_{t=1}^{n} f(X_t) = \frac{1}{\#\mathcal{S}} \sum_{\Gamma \in \mathcal{S}} c \exp\{-K f(\Gamma)\} f(\Gamma) = \beta$$

Usage of Monte Carlo techniques in the context of Markov chain takes one to 1953 article Equation of State Calculations by Fast Computing Machines by Nicholas Metropolis, Arianna W. Rosenbluth, Marshall Rosenbluth, Augusta H. Teller and Edward Teller.

The following idea of getting suitable transition function in our context is a simplified version of the same. See : `https://en.wikipedia.org/wiki/Marshall_Rosenbluth` **for some interesting commentary.**

Let us start with $p_{\Gamma,\Lambda}$ that was described earlier (doubly stochastic transition matrix) and let $\alpha(\Gamma,\Lambda) = \min\left\{1, \frac{\pi^*(\Lambda)}{\pi^*(\Gamma)}\right\}$ and

$$q_{\Gamma,\Lambda} = \begin{cases} p_{\Gamma,\Lambda}\alpha(\Gamma,\Lambda) & \text{if } \Gamma \neq \Lambda \\ 1 - \sum_{\Gamma \neq \Lambda} p_{\Gamma,\Lambda} & \text{if } \Gamma = \Lambda \end{cases}$$

For adjacent configurations if $\pi^*(\Lambda) \leq \pi^*(\Gamma)$, then

$$q_{\Gamma,\Lambda} = p_{\Gamma,\Lambda}\frac{\pi^*(\Lambda)}{\pi^*(\Gamma)}$$

and

$$q_{\Lambda,\Gamma} = p_{\Lambda,\Gamma}$$

and using $p_{\Lambda,\Gamma} = p_{\Gamma,\Lambda}$, it follows that

$$\pi^*(\Gamma)q_{\Gamma,\Lambda} = \pi^*(\Lambda)q_{\Gamma,\Lambda}$$

Using symmetry the same is true even when $\pi^*(\Lambda) \geq \pi^*(\Gamma)$.

Thus $q_{\Gamma,\Lambda}$ is an irreducible aperiodic transition probability function and that $\pi^*(\Gamma)q_{\Gamma,\Lambda} = \pi^*(\Lambda)q_{\Lambda,\Gamma}$. Hence

$$\sum_{\Gamma \in \mathcal{S}} \pi^*(\Gamma)q_{\Gamma,\Lambda} = \pi^*(\Lambda).$$

So if we can (easily) simulate Markov Chain $\{X_t : t \geq 0\}$ with transition probability function $\{q_{\Gamma,\Lambda}\}$.

We have seen how to simulate samples from $\{p_{\Gamma,\Lambda}\}$. How do we simulate samples from $\{q_{\Gamma,\Lambda}\}$?

The idea goes back to von Neumann
(perhaps used in project Manhattan!).

It is known as rejection sampling method.

John von Neumann, "Various techniques used in connection with random digits" (summary written by George E. Forsythe), pp. 36-38 of Monte Carlo Method, [U. s.] National Bureau of Standards, Applied Mathematics Series, vol. 12 (1951). Reprinted in John von Neumann, Collected Works, vol. 5, pp. 768-770, Pergamon Press, 1963.

In a technical report written by George E. Forsythe wrote

The author presents a generalisation he worked out in 1950 of von Neumann's method of generating random samples from the exponential distribution by comparisons of uniform random numbers on (0,l).

This technique called Rejection sampling is due to von Neumann (1950). Since he had been working with the Manhattan project in the 1940s, and some of these ideas may be from that project.

In the early '50s, an idea similar to that of von Neumann in the context of finite samples was introduced by D B Lahiri (at Indian Statistical Institute) to come up with probability proportional to size (pps) method of sampling.

We have seen that we can simulate Markov chain with transition probabilities $p_{\Gamma,\Lambda}$. So von Neumann's recipe is: given $X_t = \Gamma$, generate a sample Λ with probability $p_{\Gamma,\Lambda}$ and accept the proposal (of moving from Γ to Λ with probability $\alpha(\Gamma, \Lambda)$, and staying put at Γ if the proposal is rejected.

This again is easy to implement (note that in our chess board example,

$$\alpha(\Gamma, \Lambda) = \min\{1, \tfrac{f(\Lambda)}{f(\Gamma)}\}$$

and does not need any information about the class of feasible configurations.

This algorithm is known as the Metropolis algorithm for simulating a Markov Chain whose unique invariant distribution is proportional to a given function f on the state space. One does NOT need to know the constant of proportionality.

All we need is a good transition probability function that is irreducible and aperiodic on the state space that is reversible ($p_{ij} = p_{ji}$).

Hastings proposed a modification.

Suppose one has a good algorithm for a Markov chain with transition probability function $\{p_{ij} : \ i, j \in S\}$ that is easy to simulate and we with to obtain a transition probability function $\{q_{ij} : i, j \in S\}$ that has $\{h(i) : i \in S\}$ as the stationary distribution, let

$$q_{ij} = p_{ij} \min\left\{1, \frac{h(j)p_{ji}}{h(i)p_{ij}}\right\} \quad \text{if} \ \ i \neq j$$

and $q_{ii} = 1 - \sum_{j \neq q} q_{ij}$.

It can be checked that

$$h(i)q_{ij} = h(j)q_{ji} \quad \forall i, j \in S$$

and thus

$$\sum_{i \in S} h(i)q_{ij} = h(j).$$

As in the Metropolis algorithm, if we can simulate from $\{p_{ij} : j \in S\}$ then using rejection sampling we can simulate from $\{q_{ij} : j \in S\}$.

How to choose initial conditions X_0?

One sample or several samples ??

The same result works in $S = \mathbb{R}$ and $S = \mathbb{R}^d$.

Aim: To approximate

$$\int_S \phi(y)h(y)d\mu(y)$$

where $h(y) = ch_1(y)$ is a density and ϕ is a given function.

Suppose that the constant c is not known, but only h_1 is known.

Suppose $f(x,y)$ is density of a Markov chain on S,
such that $f(x,y) = f(y,x)$ and for $A \in \mathcal{B}(S)$ one has

$$P(X_{t+1} \in A \mid X_t = x) = \int_A f(x,y) d\mu(y)$$

We define a transition function Q as follows: Let

$$\alpha(x,y) = \min\{1, \tfrac{h(y)}{h(x)}\} = \min\{1, \tfrac{h_1(y)}{h_1(x)}\}$$

and for $A \in \mathcal{B}(S)$, $x \notin A$

$$Q(X_{t+1} \in A \mid X_t = x) = \int_A f(x,y)\alpha(x,y) d\mu(y)$$

$$Q(X_{t+1} = x \mid X_t = x) = 1 - \int_{S-\{x\}} f(x,t)\alpha(x,t) d\mu(t).$$

Once we have a efficient algorithm to simulate $X_{t+1} = y$ given $X_t = x$, we treat the move as proposal, and accept the proposal (of moving from x to y with probability $\alpha(x, y)$, and staying put at x if the proposal is rejected.

The density $f(x, y)$ in turn could be $f(x, y) = \phi(y - x)$ where ϕ is a suitable density with mean 0.

Examples

1: ϕ being Gaussian distribution with mean 0, variance σ^2.

2: ϕ being Laplace distribution with mean 0 and scale β.

3: ϕ being Uniform distribution on $(-a, a)$, $a > 0$.

Likewise we can have a version of Hastings algorithm, called Metropolis-Hastings algorithm:

Suppose $f(x, y)$ is density of a Markov chain

$\{X_t : t \geq 0\}$ on S,

so that for $A \in \mathcal{B}(S)$ one has

$$P(X_{t+1} \in A \mid X_t = x) = \int_A f(x, y) d\mu(y).$$

We assume that such an $\{X_t : t \geq 0\}$ on S is easy to simulate

We define a transition function Q as follows: Let

$$\alpha(x,y) = \min\{1, \tfrac{h(y)f(y,x)}{h(x)f(x,y)}\} = \min\{1, \tfrac{h_1(y)f(y,x)}{h_1(x)f(x,y)}\}$$

and for $A \in \mathcal{B}(S)$, $x \notin A$

$$Q(X_{t+1} \in A \mid X_t = x) = \int_A f(x,y)\alpha(x,y)d\mu(y)$$

$$Q(X_{t+1} = x \mid X_t = x) = 1 - \int_{S-\{x\}} f(x,t)\alpha(x,t)d\mu(t).$$

For the Metropolis-Hastings algorithm, we do not need to assume that $f(x, y)$ is symmetric. For example, we can even take $f(x, y) = \phi(y)$!

We could even take a combination of two algorithms - at each step, we use one algorithm with probability p and the other with probability $1 - p$.