CHAPTER 7: Markov Chain Monte Carlo Methods

(a) Metropolis/Hastings

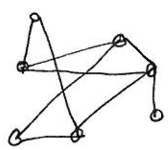
Suppose that we wish to generate a random variable X, where $\mathbb{P}(X=i)=\pi_i$ is not known exactly, but only up to a normalization constant (or factor). That is, we know $b(i)$; $i\in S$, but calculating

$$K=\sum_{i\in S} b(i) \qquad (1)$$

is a very difficult numerical task (large combinatorial problem). Here,

$$\pi_i=\mathbb{P}(X=i)=\frac{b(i)}{K}.$$

Example : Telecommunications or LAN network : connects nodes via physical links. Each link has a different failure probability



original network (physical)
sub-trees represent working links

Reliability problem: evaluate the probability that all nodes are connected, knowing the individual link failure joint probabilities. Banks as example : sensitive data, storage systems, distributed operations through access points (ATM's), etc. (Other examples in genetic information, national security, crime labs...).

$$\Rightarrow R=\sum_{V\in\mathcal{V}}\mathbb{P}(V \text{ is a connected graph}), \quad \mathcal{V}: \text{set of all subgraphs.}$$

Bruteforce approach:

- Enumerate all possible "up-down" link scenarios,
- For each one, calculate if it is ~~a subtree~~ connected.

(if $\exists$ a subtree).

Instead, suppose that we can generate all connected subtrees with uniform probability. Then we generate the corresponding link variables $\xi_e\in\{0,1\}$. If at least one of them is not working ($\xi_e=0$) then $X_n=0$ for that sample. Otherwise $X_n=1$.

$\Rightarrow$ How to generate the sub-trees with uniform probability? We only know $b(i)=K$ a constant.

IDEA: Build a succesive algorithm that will be a Markov chain $\{X_n\}$ such that π_i are the limiting probabilities. Then for any bounded function h we can estimate $\mathbb{E}h(X)$ using samples:

$$\mathbb{E}(h(X))=\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}h(X_n).$$

Thm: Let $Q=\{q_{ij}\}$ be an irreducible matrix, $i,j\in S$. Define a Markov chain $\{X_n\}$ through its transition probabilities:

$$P_{ij}=\begin{cases} q_{ij}\,\alpha_{ij} & i\neq j\\ q_{ii}+\sum_{k\neq i} q_{ik}(1-\alpha_{ik}) & i=j\end{cases}$$

where $\alpha_{ij}=\min\left(\frac{b(j)\,q_{ji}}{b(i)\,q_{ij}},1\right)$. Then $\{X_n\}$ is an ergodic MC with limiting probabilities $\pi_i=\frac{b(i)}{K}$,

<u>Proof</u>: Using the theorem for reversible Markov chains, if we can verify that $\forall i,j \in S$

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \forall i \neq j, \qquad (2)$$

then the claim follows, identifying $b(i) = K\pi_i$. Now given $i \neq j \in S$, we have two possibilities:

$$\alpha_{ij} = \frac{b(j)q_{ji}}{b(i)q_{ij}} \quad \text{and} \quad \alpha_{ji} = 1, \text{ or}$$

$$\alpha_{ij} = 1 \quad \text{and} \quad \alpha_{ji} = \frac{b(i)q_{ij}}{b(j)q_{ji}}.$$

Suppose wlog that $\alpha_{ij} = \frac{b(j)q_{ji}}{b(i)q_{ij}} \leq 1$, then by definition:

$$P_{ij} = q_{ij}\alpha_{ij} = \frac{\pi_j}{\pi_i} q_{ji} \Rightarrow \pi_i P_{ij} = \pi_j q_{ji}.$$

and $P_{ji} = q_{ji}\alpha_{ji} = q_{ji}$. Thus (2) is verified. QED.

ALGORITHM:

$i = X_n$

Generate $j \sim Q_i$.

Generate $U_{n+1} \sim U(0,1) \perp\!\!\!\perp j$

If $U_n \leq \alpha_{ij} \Rightarrow X_{n+1} = j$

Else $X_{n+1} = i$

The MCMC methods have the general form of an acceptance/rejection test with state-dependency. Today, many "search" algorithms have the general structure of Markov chains.

<u>(b) The Gibbs Sampler</u>.

Generalizes M-H to vectors of random variables.

Example 4.39 p.262-263 Ross and 10a. p. 250 Ross (8).

<u>Example</u>: $\{1, \dots n\}$ a given set of numbers, and

$$\mathcal{P} = \{ (x_1, \dots x_n) \text{ a permutation} : \sum_j j x_j > a \},$$

where $a > 0$ is a given number. Goal: generate a uniform distribution on the set $\mathcal{P}$.

Define the concept of "neighborhood" as follows: if $(x_1, \dots x_n)$ is a given vector, then a neighbor is another vector obtained by swapping two elements $i \neq j$, that is:

$$(x_1, \dots x_i, \dots x_j, \dots x_n) \text{ and}$$
$$(x_1, \dots x_j, \dots x_i, \dots x_n)$$

are neighbors

swapping values

$(y_1, \dots y_n)$ is a neighbor of $(x_1, \dots x_n)$ if $\exists\, j,k \in \{1, \dots n\}$, $j \neq k$ such that $y_i = x_i \ \forall i \neq \{j,k\}$, $y_k = x_j$ and $y_j = x_k$.

We may want to use, for example, a uniform probability on the neighborhoods to generate the candidate:

$$q(x,y) = \frac{1}{|N(x)|} \mathbb{1}(y \in N(x))$$

[In example, what is $|N(x)|$? $x \in P \Rightarrow \leq \binom{n}{2}$]

and then use:

$$\alpha(x,y) = \min\left(\frac{|N(x)|}{|N(y)|}, 1\right).$$

General structure:

- A neighborhood of each possible state,
- A distribution Q for the candidate,
- The acceptance/rejection test or probability.

A particular case of application of the vectorial version of the M-H algorithm above is when the conditional probabilities:

$$\mathbb{P}(X_i = x_i \mid X_j = x_j ; j \neq i) = p(x \mid \bar{x}_i) \quad (3)$$

are known exactly, even though $\mathbb{P}(X = x) = \pi_x$ is not known.
[Examples in Ross (S) and SGS paper.]

$\bar{x}_i = (x_j ; j \neq i)$.

In this case, a neighborhood of $(x_1, \dots x_n)$ is defined by: $N(x) = \{ y : (y_j = x_j ; j \neq i) ; i = 1, \dots n \}$

That is, only one component of y is different from the corresponding one in x. Application of MH uses the distribution for candidates given by:

$$q(x,y) = \frac{1}{n} \sum_{i=1}^{n} p(y_i \mid \bar{x}_i) \mathbb{1}(\bar{y}_j = \bar{x}_j ; j \neq i)$$

It corresponds to choosing a random index $i \in \{1, \dots n\}$ uniformly and generating only the component X_i conditional on $\bar{x}_i$ in (3). Because it shapes the distribution of the candidate exactly to fit the conditional distribution of X (the target), it turns out that there is no rejection in the algorithm.

Exercise: show that $\alpha(x,y) = 1$ for this algorithm (p.251 R-S).

The above algorithm is called the Gibbs sampler.
[Examples in book p. 254-262 Ross-S].

DISCRETE OPTIMIZATION WITH MCMC

(c) Simulated Annealing

S is a finite but probably very large set, say $S = \{1, \dots m\}$.
Let $f(x)$ be a cost associated with state $x \in S$.
We wish to find the optimal "design" or "configuration" that minimizes the cost, that is:

$$f^* = \min_{x \in S} f(x).$$

Let $\mathcal{M} = \{x \in S : f(x) = f^*\}$ be the optimal set.

The parameter T is called the "temperature" and we define $\lambda = 1/T$ as the algorithm's parameter.

Define the probability:

$$P_\lambda(x) = \frac{e^{-\lambda f(x)}}{\sum_{x\in S} e^{-\lambda f(x)}} = \frac{e^{-\lambda[f(x)-f^*]}}{|\mathcal{M}| + \sum_{x'\notin \mathcal{M}} e^{-\lambda[f(x')-f^*]}}.$$

Notice that $f(x)-f^* > 0 \ \forall x \notin \mathcal{M}$, thus as $\lambda \to +\infty$

$P_\lambda(x) \to 0$ for all $x \notin \mathcal{M}$, and therefore the limiting probability (as λ increases) is concentrated on the optimal set. Mathematically, if $X_\lambda \sim P_\lambda(\cdot)$ then as $\lambda \to \infty$

$$X_\lambda \Rightarrow X_\infty \quad \text{(convergence in distribution)}$$

where $\mathbb{P}(X_\infty \notin \mathcal{M}) = 0$.

The idea is to use MCMC to produce a Markov chain $\{X_n(\lambda)\}$ with limiting probabilities $P_\lambda(\cdot)$.

Remark: λ is a "design parameter" chosen by the programmer, and it is assumed that for any $x \in S$, the value $f(x)$ can be evaluated or observed (from a simulation, an observation, or an execution of a computation). However, because $|S|$ is very large, the calculation of the normalization factor $K = \sum_{x\in S} e^{-\lambda f(x)}$ may be impossible or impractical. Here we identify $b(x) = e^{-\lambda f(x)}$.

The 'neighborhoods' of any element $x \in S$ can be defined in any convenient manner, as long as they connect all the state space (see proposition below for precise condition).

Example: if the state space contains vectors such as the buffer occupancies in large computer networks, then a neighbor may be any other vector that differs in only one of the component values.

Let: $$q(x,y) = \frac{1}{|N(x)|} \mathbb{1}(y \in N(x)).$$

Proposition: The matrix Q is irreducible iff $\forall x, y \in S$ there is a sequence of states $x = i_1, i_2, \ldots, i_m = y$, with $i_{k+1} \in N(i_k)$. } called the 'reachability property'

Thm: If the neighborhoods satisfy the 'reachability' property, using:

$$\alpha^{(\lambda)}_{x,y} = \min\left(\frac{e^{-\lambda f(y)}}{e^{-\lambda f(x)}} \frac{|N(x)|}{|N(y)|}, 1\right)$$

to build a M-H Markov chain $\{X_n(\lambda)\}$, then this chain is ergodic and it has limit probabilities $P_\lambda(\cdot)$.

In the algorithm, if the current state $X_n = i$, and assuming that $|N(i)| = \text{cte}$, then:

- j is uniformly chosen in $N(i)$
- ($\alpha_{ij}=1$) if $f(j) < f(i) \Rightarrow$ "move to j" ($X_{n+1} = j$)
- if $f(j) \geq f(i) \Rightarrow$ move to j w.p. $e^{-\lambda(f(j)-f(i))} < 1$

(interpret: why do we move?)

PROBLEM: The limiting probabilities ~~are~~ do not ensure that the algorithm will converge to an optimal value, even if this has "large" probability.

SOLUTION: (i) Use sequentially increasing parameters $\lambda_n \to \infty$ ($T_n \to 0$ is associated with cooling temperatures in the annealing process).

(ii) Main questions: how fast should λ_n increase? Should one use a bi-level approach or a two-time scale approach?

Bilevel: For each λ_n, find $\lim_{k\to\infty} \{\mathbb{P}(X_k(\lambda))\}$ by approximation (when to stop the simulation?)

Two-time scale: Use a non-homogeneous MC model, changing the candidate probabilities at each iteration:
$$P_{ij;n} = \begin{cases} q_{ij}\,\alpha_{ij}(\lambda_n) & i \neq j \\ q_{ii} + \sum_{k\neq i} q_{ik}(1-\alpha_{ik}(\lambda_n)) \end{cases}$$

We will study the question on ~~the~~ when to stop a simulation in our chapter on "output analysis". For the two-time scale problem, techniques such as weak ergodicity conditions and stochastic approximation have been used to establish that if $\lambda_n \le C\log(1+n)$ then X_n converges in distribution to a limit rv with support on the optimal set $\mathcal{M}$.

Comments: very popular algorithm 80's and 90's, but "slow"?

(d) Stochastic Ruler

Suppose that, given a "design" or choice $x \in S$ (a very large set), the cost function $f(x)$ cannot be computed analytically, and can only be observed with noise. Specifically, $\exists$ rv. ξ_x on a space $(\Omega, \mathcal{F}, \mathbb{P})$ such that
$$f(x) = \mathbb{E}(h(x,\xi_x)), \in (a,b).$$

To simplify notation, we will assume that given a value x, an observation $\hat{f}(x) = h(x,\xi_x)$ is made, and that it is statistically independent of previous observations.

ALGORITHM: $i = X_n$ is current state

- Generate a candidate $j \in N(i)$ (neighborhood) with distribution $Q(i,\cdot)$
- For $k = 1, \dots, M_n$ ($M_n \to \infty$)
  - Generate $\hat{f}^{(k)}(j)$
  - Generate $R^{(k)} \sim U(a,b)$ "stochastic ruler"
  - If $\hat{f}^{(k)}(j) > R^{(k)} \Rightarrow$ STOP & $X_{n+1} = X_n$.

    else continue and set $X_{n+1} = j$.

Consecutive values of $\{X_n\}$ are estimates of the optimal value x^*, ~~and~~ where $f(x^*) \le f(x)\ \forall x \in S$.

Point generated j (w.p. $Q(i,j)$) is accepted only when all the observations $\{h(j,\xi_j^{(k)}), k=1,\dots M_n\}$ ~~are~~ satisfy $h(j,\xi_j^{(k)}) \le R^k$. The acceptance prob. is:

$$\mathbb{P}(X_{n+1}=j \mid X_n=i) = Q(i,j) \prod_{k=1}^{M_n} \mathbb{P}(h(j,\xi_j^{(k)}) \le R^{(k)})$$

Use: $\mathbb{P}(h(j,\xi_j^{(k)}) \geq R^{(k)}) = \mathbb{E}\left(\mathbb{P}(R^{(k)} \le h(j,\xi_j^{(k)}) \mid h(j,\xi_j^{(k)})\right)$

$$= \mathbb{E}\left(\frac{h(j,\xi_j^{(k)})-a}{b-a}\right) = \frac{f(j)-a}{b-a}.$$

$\Rightarrow$ smaller values of $f(j)$ have higher acceptance probabilities.

Let $P(j) = 1 - \dfrac{f(j)-a}{b-a}$, then

$$\mathbb{P}(X_{n+1}=j \mid X_n=i) = Q(i,j)(P(j))^{M_n},$$

which is largest when $f(j)$ is closest to a (minimal possible value). [See analysis in AlfAnd 1997].

Remark: if $P(i) \geq P(j) \Rightarrow f(i) \le f(j)$.

For any value $X_n \in S$, we have here:

$$\mathbb{P}(X_{n+1} \neq x^* \mid X_n) \approx (1-\rho)^{M_n} \to 0$$

where $\rho = \min\limits_{x \neq x^*} \dfrac{f(x)-a}{b-a} > 0.$

therefore, similarly to simulated annealing, $M_n \to 0$ at a certain rate to ensure convergence.

Result: As $M_n \to \infty$ with $M_n \approx O(\ln n)$, $X_n \xrightarrow{P} x^*$

Generalizations: other supports, accelerating convergence, choice of neighborhoods ...

$\to$ (*) p. 7

(e) Stochastic Comparisons

The set-up is as in (d), where observations are noisy but unbiased.

~~Generate~~

ALGORITHM: $i = X_n$, $\bar{f}(i)$ current estimate of $f(i)$

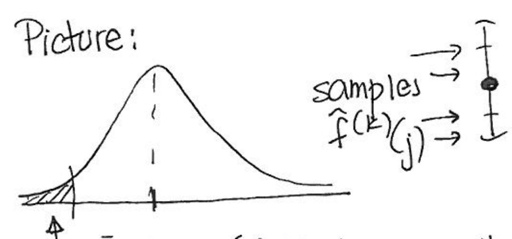
- Generate $j \sim Q(i,\cdot)$, $j \in N(i)$
- For $k = 1, \ldots M_n$
  - Generate $\hat{f}^{(k)}(j)$
  - If $(\hat{f}^{(k)}(j) > \bar{f}(i)) \Rightarrow X_{n+1} = i$

    else continue and set:

    $X_{n+1} = j$

    $\bar{f}(j) = \dfrac{1}{M_n}\sum\limits_{k=1}^{M_n} \hat{f}^{(k)}(j)$ sample avg.

Picture:



$\bar{f}(i)$ is current avg.

$\bar{f}(i) < f^{(k)}(j)$ has small probability if $f(i) > f(j)$.

$\mathbb{P}(f^{(k)}(j) > \bar{f}(i)) =$

$\mathbb{P}(f(j) + \bar{w}(j) > f(i))$

$= \mathbb{P}(\bar{w}(j) > f(i) - f(j))$, $\mathbb{E}\bar{w} = 0$.

Here, $\mathbb{P}(X_{n+1}=j \mid X_n=i, f(i)<f(j)) \approx [p(i,j)]^{M_n}$

where $P(i,j) < 1$, so that acceptance $\to 0$ as $M_n \to \infty$.

General Random Search Methods

The structure of these algorithms is always of the form:

- Define neighborhood structure satisfying reachability.

Trade-off between simple structures and small neighborhoods and overall speed: the neighborhoods determine the "exploration" capabilities. Try to define "clever" structures.

- Acceptance/rejection criteria in terms of function evaluations. Because of noise in observations, the amount of samples determines the "exploitation" requirements: reduce noise $\Leftrightarrow$ increase computational effort. It is often the case that for off-line routines the program keeps a "best candidate" solution using cummulative averages.

- Adapting the exploration density: genetic algorithms, ants, bee swarms, cross-entropy methods and other popular heuristics.

- Convergence analysis (VERY IMPORTANT, OPEN QUESTION FOR MANY MACHINE LEARNING METHODS)

(*) Example: stochastic ruler with constant M. This procedure will yield positive probability to be away from x^*, but best candidate may still converge to x^*, as shown:

Thm: (AlfAnd 1991 p. 354)

(a1) $\{N(i), i \in S\}$ satisfy the reachability condition and that $\forall i \in S$, $i \notin N(i)$.

(a2) If $P(i) \geq P(j) \Rightarrow f(i) \leq f(j)$

Let: $V_n(i) = \sum_{k=1}^{n} \mathbb{1}(X_n = i)$ be the total number of visits to state i up to iteration n, and define

$$X_n^* = \begin{cases} X_n & \text{if } V_n(X_n) > V_n(X_n^*) \\ X_n^* & \text{otherwise} \end{cases}$$

If $Q(i, \cdot)$ is the uniform sampling probability on $N(i)$ $\forall i \in S$, then $X_n^* \to x^*$ w.p.1.

[Andradottir 1996]

$$\min_{\theta \in S} f(\theta), \qquad f(\theta) = \mathbb{E}(X_n(\theta))$$

$\uparrow$ some simulation experiment

$|S| = K$

S^*: set of optimal solutions.

<u>Assumption 1</u>: $\exists$ rv $Y^{(i \to j)}$:

$$i \in S^*,\ j \notin S^* \Rightarrow \mathbb{P}(Y^{(j \to i)} > 0) > \mathbb{P}(Y^{(i \to j)} > 0)$$

$$n \neq i,j \Rightarrow \mathbb{P}(Y^{(n \to i)} > 0) > \mathbb{P}(Y^{(n \to j)} > 0)$$

[interpret: "moves" in right direction have larger probs].

<u>Example</u>: stochastic comparisons may use $Y^{(i \to j)} = X_n(i) - X_n(j)$, w. independent sampling.

- Generate θ_n' (uniform on $S \setminus \theta_n$)
- Generate an observation of the test $Y^{\theta_n \to \theta_n'}$ and call it R_m. If $R_m > 0 \Rightarrow$ accept: $\theta_{n+1} = \theta_n'$, otherwise $\theta_{n+1} = \theta_n$.
- Count the visits: $N_{n+1}(\theta_{n+1}) = N_n(\theta_{n+1}) + 1$, $N_{n+1}(\cdot) = N_n(\cdot)$ for others.
- Candidate: if $N_{n+1}(\theta_{n+1}) > N_{n+1}(\Theta_n^*) \Rightarrow \Theta_{n+1}^* = \theta_{n+1}$.

(Neighborhood structure can be generalized.)

<u>Theorem</u>: Under assumption 1, $\{\theta_n\}$ converges:

$\theta_n \to \theta^* \in S^*$ w.p.1.

<u>Proof</u>:

(1) $\{\theta_n\}$ is a MC:

$$p(i,j) = \frac{1}{K-1} \mathbb{P}(Y^{(i \to j)} > 0), \quad i \neq j$$

$$p(i,i) = 1 - \sum_j \mathbb{P}(i,j) = 1 - \frac{1}{K-1} \sum_j \mathbb{P}(Y^{(i \to j)} > 0) = \frac{1}{K-1} \sum_j (1 - \mathbb{P}(Y^{(i \to j)} > 0)) = \frac{1}{K-1} \sum_j \mathbb{P}(Y^{(i \to j)} \leq 0)$$

(assumes indep. samples $\{R_n\}$ of the test).

Suppose the chain is irreducible w. stat prob π.

$$\pi_j = \sum_{i \in S} \pi_i \, p(i,j).$$

By assumption 1 + algebra, it is shown that if $i \in S^*$, $j \notin S^* \Rightarrow \pi_i > \pi_j$. This shows that $\operatorname{argmax} \pi_j \in S^*$.

$v_j = \mathbb{E}(\#\text{iterations between visits to } j) = 1/\pi_j$.

Notice that

$$\frac{N_n(\theta)}{n} \to \pi_\theta \text{ a.s.}$$

$\exists$ null set A^c such that $\forall \omega \in A$

$$\frac{N_n(\theta,\omega)}{n} \to \pi_\theta$$

then, by definition, for each $\omega \in A$

$$\Theta_n^*(\omega) = \arg\max \{N_n(\theta,\omega)\}$$

$$= \arg\max \left\{\frac{N_n(\theta,\omega)}{n}\right\}$$

Because of a.s. convergence and the fact that

$\pi_i - \pi_j \geq \delta > 0 \quad \forall i \in S^*, j \notin S$, then $\exists$

$m(\omega) : \forall n \geq m(\omega), \Theta_n^*(\omega) \in S^*$. QED.

- How to construct the test?
- Assumption 1 may not be verifiable for processes, but some limit.

$$\min_{\theta \in S} f(\theta) = \mathbb{E}[g(\theta, X_1(\theta), \ldots X_\tau(\theta))]$$

τ can be random stopping time.

Theorem 3.3 in And. p. 522 (1996)

<u>Example 1:</u> buffer allocation in routing network.
pages 2-3 Leyuan Shi. + page 22.

<u>Example 2:</u> stochastic travelling salesman with D items to sell.
Find route $(r_1, \ldots, r_m)$ (permutation of $(1,..m)$) with optimal "cost".

Time of travel $t(i,j)$ is random

For each i, demand of good and price vary:

$d(i)$, $p(i)$ are random

What is the cost of route?:

$$c \sum_i t(r_i, r_{i+1}) - \sum_{i=1}^{\tau} d(r_i)\, p(r_i)$$

where $\tau = \min(i : \sum_{j=1}^{i} d(r_j) = D)$.

May seek to minimise $\mathbb{E}(\text{cost})$.