Chapter 7.4. Selected Topics: Monte Carlo Methods

Markov chains Monte Carlo

One of the most relevant applications of Markov chains is sampling (S-valued) Random Variables. Given a target measure $m \in \mathcal{P}(S)$, one is interested in sampling a random variable X with distribution m, namely $\mathbb{P}(X = x) = m_x$ for all $x \in S$. A possible strategy is to run a Markov chain (ideally irreducible, aperiodic) on S, with invariant measure m. As one simulates the chain for a long time T, the law of X_T will converge exponentially fast to the target measure m. Of course it is important, and highly non-trivial, to estimate accurately the speed of convergence of the law of X_T. Most importantly, given the target measure m, to *design* some transition probabilities that will guarantee a fast convergence and a low computational cost. The latter property can be formalized mathematically in many ways (e.g. avoiding the computation of some complex quantities, the number of calls of m_x per step, or simply the simulation algorithm being nearest-neighbor w.r.t. some natural structure on the graph where the target measure m is defined), none of them fully satisfying.

Example 0.1 (Simulated Annealing)**.** Suppose one wants to find the global minimum of a *cost function* $U : S \to \mathbb{R}$. In many applications (e.g., combinatorial optimization), S is very large and U has many local minima, making local or deterministic search algorithms ineffective.

The idea of simulated annealing is to turn this optimization problem into a sampling problem. For a parameter $\beta > 0$ (interpreted as the inverse temperature), define the **Gibbs measure**:

$$m_x^\beta = \frac{1}{Z^\beta} e^{-\beta U(x)}, \qquad Z^\beta = \sum_{y \in S} e^{-\beta U(y)}$$

As $\beta \to \infty$ (temperature goes to zero), the measure m^β concentrates on the set of global minima of U. If we can sample from m^β for very large β, we find the minimum with large probability (the probability to miss it is exponentially small in β).

However, direct sampling is impractical, e.g. due to the constant Z^β (computing Z^β implies computing all the $U(x)$). Instead, we run a Markov chain designed to have m^β as its invariant measure. Let us assume that S has a graph structure, with degree bounded by D (that is, no point has more than D neighbors). This is motivated by the fact that, although $|S|$ is large in many cases, it has a (connected, undirected) graph structure with a *small* degree (see Example 0.2 below; or consider the case of a large box $S = \{-N, \dots, N\}^d$ in $\mathbb{Z}^d$, so $|S| = (2N+1)^d$ but $D = 2d$). We then take

$$p_{x,y}^\beta = \begin{cases} e^{-\beta(U(y)-U(x))^+}/D & \text{if } x, y \text{ are neighbors,} \\ 1 - \sum_{z \text{ neighbor of } x} e^{-\beta(U(z)-U(x))^+}/D & \text{if } y = x, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where a^+ denotes the positive part of a real number a.

This choice corresponds, at a given step when $X_t = x$, to choose with probability $1/D$ a given *candidate* neighbor y of x. If $U(y) \le U(x)$, then $X_{t+1} = y$. If $U(y) > U(x)$, then with probability $e^{-\beta(U(y)-U(x))}$ we set $X_{t+1} = y$, and otherwise one does not move, $X_{t+1} = x$. The idea is that this procedure, on the one hand favors jumping toward

smaller values of U, and on the other hand allows (even if with small probability) "escaping valleys" around local minimizers of U. The invariance of m for this random dynamics is a consequence of the more general Proposition 0.1.

In practice, in the so-called **simulated annealing** algorithm, one simulates this chain with a slowly increasing β (so $\beta = \beta_t$) as the time t increases, theoretically allowing the chain to escape local minima and settle in the global minimum. As increasing β means decreasing the temperature over time, the name of this influential technique comes from a heat treatment of metals.

Example 0.2 (Traveling salesman)**.** Consider N cities labeled $1, \dots, N$, and a matrix $(c_{i,j})_{1\le i,j\le N}$ representing the cost of traveling between city i and city j. A traveling salesman has to visit all N towns and come back home, and has to decide the optimal order of towns to visit. This means we are looking for a permutation $\sigma \in \mathcal{S}_N$ that minimizes the total cost.

Here, the state space $S = \mathcal{S}_N$ is the set of all permutations of the N cities. The cost function $U : S \to \mathbb{R}$ is the total cost of the tour (we understand sums $\pmod N$, so $N + 1 = 1$):

$$U(\sigma) = \sum_{i=1}^{N} c_{\sigma(i),\sigma(i+1)}$$

To apply the method in Example 0.1, we must define a graph structure on S. Let us try different strategies.

a. Two permutations are connected if one can be obtained from the other by **swapping** two indices. If we denote by $\pi^{i,j}\sigma$ the transposition swapping the entries at position i and j, the neighbors of σ are the $D = \binom{N}{2}$ permutations $(\pi^{i,j}\sigma)_{i<j}$.
b. Two permutations are connected if one can be obtained from the other by swapping two *consecutive* indices. This means that the neighbors of σ are all the permutations obtained from σ with an *adjacent* transposition, namely the $D = N$ permutations $(\pi^{i,i+1}\sigma)_i$.
c. If $c_{i,j} = c_{j,i}$ is **symmetric**, two permutations are connected if one can be obtained from the other by *reversing* the order of visit of a sub-segment of indices. If we denote by $\epsilon^{i,j}\sigma$ the permutation obtained by reversing the path between index i and j, the neighbors of σ are the $D = \binom{N}{2}$ permutations $(\epsilon^{i,j}\sigma)_{i<j}$.

Now, as in Example 0.1, we fix a large constant $\beta > 0$ (which governs the probability of accepting "uphill" moves), and we simulate a Markov chain with transition probabilities given by Equation 1. A crucial computational advantage of the local dynamics a.,b.,c.., is that we do not need to ever calculate the full cost $U(\sigma)$, but only the difference of $U(\cdot)$ between neighbor permutations. This is easily computed (see Figure 1) as

$$\delta U := \begin{cases} U(\pi^{i,j}\sigma) - U(\sigma) = (c_{\sigma(i-1),\sigma(j)} + c_{\sigma(j),\sigma(i+1)} + c_{\sigma(j-1),\sigma(i)} + c_{\sigma(i),\sigma(j+1)}) - (c_{\sigma(i-1),\sigma(i)} + c_{\sigma(i),\sigma(i+1)} + c_{\sigma(j-1} \\ U(\pi^{i,i+1}\sigma) - U(\sigma) = (c_{\sigma(i-1),\sigma(i+1)} + c_{\sigma(i+1),\sigma(i)} + c_{\sigma(i),\sigma(i+2)}) - (c_{\sigma(i-1),\sigma(i)} + c_{\sigma(i),\sigma(i+1)} + c_{\sigma(i+1),\sigma(i+2)}) \\ U(\epsilon^{i,j}\sigma) - U(\sigma) = (c_{\sigma(i-1),\sigma(j)} + c_{\sigma(i),\sigma(j+1)}) - (c_{\sigma(i-1),\sigma(i)} + c_{\sigma(j),\sigma(j+1)}) \end{cases} \tag{2}$$

Thus to evaluate Equation 1, we never need to compute Z^β or even U, just a combination of respectively $8, 6, 4$ cost terms (for cases a.,b.,c.) (regardless of N) . The algorithm is then

1. Start with an arbitrary permutation, e.g. the identity or a random one.
2. Pick two random indices i, j (for the dynamics a.), or one random index i (for dynamics b., as $j = i + 1$ in this case).
3. Compute the value δU in Equation 2.
4. If $\delta U \le 0$, then perform the step (namely swap i and j in case a., swap i and $i + 1$ in case b., reverse the path between i and j in case c.). If $\delta U > 0$, then perform the step with probability $e^{-\delta U}$, and do not make any change otherwise.
5. Repeat from point 2..

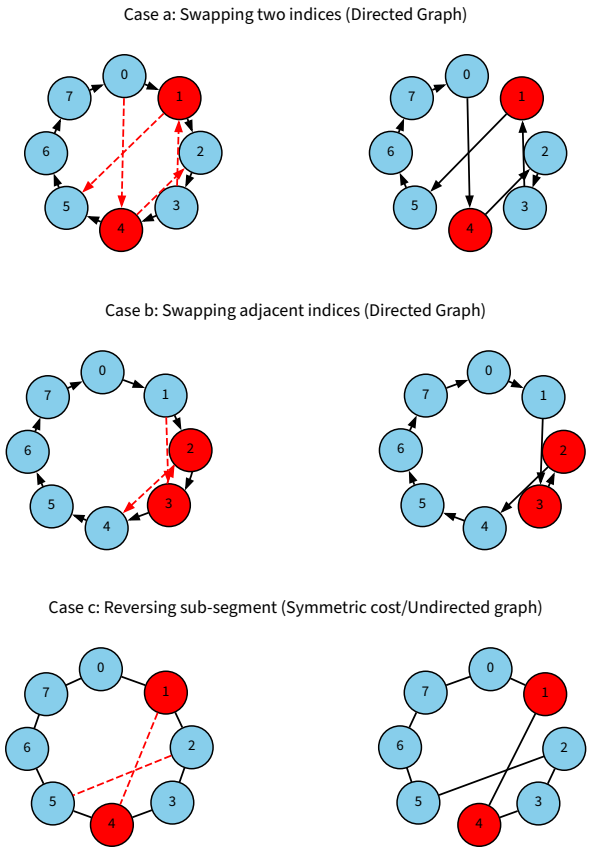

Figure 1: The different single-step transitions for the three examples of the Salesman problem.

Of course, a critical question is how many times we should repeat steps 2.-5.. This is a problem of great practical and theoretical interest. The answer critically depends on N, β, U, and the acceptance of possible errors. In many situations, one may just take "T large enough" and avoid mathematical complications.

In the examples a.,b.,c. above, it is quite clear what to expect:

- The cases a. and c. are relatively similar, each point has $N(N-1)/2$ neighbors, and they usually need less steps to visit the state space and "find" the minimizers of U. Usually c. is expected to perform better than a.. b. instead only allows N neighbors, and has a higher chance to get stuck in "local minimizers" of U (namely, to find a path of lower cost, it first needs to swap several adjacent indices where the cost *increases*, but these transitions are unlikely as β gets large). In simulations, we see that that b. is indeed the algorithm for which U decreases the slowest with iterations.
- However each single step of b. is much simpler computationally, and if we plot instead $U(X_t)$ not as the function of the number of steps, but the cpu time taken to simulate, the gap between b. and the other algorithm is much reduced.
- Ultimately, we see that increasing β as the simulation goes on is the best strategy, and certainly for b. to perform, we just need to tune $\beta \equiv \beta_t$ so that it grows, but slowly compared to a. and b..

Ultimately, the technique boils down to the following approach, called MCMC (Markov chain Monte Carlo). Given $m \in \mathcal{P}(S)$, find some "computationally optimal" transition probabilities $(p^t_{x,y})_{t\geq 0, x,y\in S}$ such that $\sum_y m_y p^t_{y,x} = m_x$, for all $t \in \mathbb{N}, x \in S$. We will focus on the time-homogeneous case, where $p^t_{x,y} \equiv p_{x,y}$ does not depend on t.

Let us give some examples, assuming (with no loss of generality) $m_x > 0$ for all $x \in S$.

1. $p_{x,y} = m_y$, the **trivial perfect simulation**, yielding a Markov chain which is just an i.i.d. sequence. In this case, the law of X_t converges to m in just one step.

2. $p_{x,y} = \varepsilon\sqrt{m_y/m_x}\mathbf{1}_{x\neq y} + (1-\varepsilon\sum_{z\neq x}\sqrt{m_z/m_x})\mathbf{1}_{x=y}$, known as the **global heat bath**. Here $0 < \varepsilon \leq \inf_{x,y}\sqrt{m_x/m_y}$, so that $p_{x,y}$ is indeed a transition probability with reversible, and thus invariant, measure m.
3. $p_{x,y} = A_{x,y}\sqrt{m_y/m_x}$, generalizing the previous examples. Here $A_{x,y}$ is a *sufficiently small* symmetric matrix, meaning
$$A_{x,y} = A_{y,x} \geq 0, \qquad x,y \in S, \qquad A_{x,x}\sqrt{m_x} = 1 - \sum_{z\neq x} S_{x,z}\sqrt{m_z}, \qquad x \in S \tag{3}$$
Clearly $m_x p_{x,y} = A_{x,y}\sqrt{m_x m_y} = m_y p_{y,x}$, so that m is reversible, and thus invariant.

In typical applications, S is **large** and m_x is a somehow *complex* measure. The first two examples above are highly impractical. In the first case, we just sample m deterministically. The instantaneous convergence however implies a very high computational cost of the single step needed. In the global heat bath, at each step we need to know the ratios $\sqrt{m_y/m_x}$ for *all* $x, y \in S$, and despite this being typically simpler to compute than m_x (see Example 0.2), it still requires the access to $|S|^2$ ratios of the invariant measure values. The third general case is more interesting (and will be further generalized below), but so far does not tell us much about a good choice of $A_{x,y}$.

Metropolis-Hastings dynamics

The problem becomes much more interesting both mathematically and practically, if we assume that some further structure is given. In particular, we assume that one is given some *original* transition probabilities $q_{x,y}$. Beware, we are not assuming that m is invariant for $q_{x,y}$ here, rather $q_{x,y}$ should be thought as deriving from some feature, for instance geometric properties, of the problem of interest. If S has a graph structure for instance, $q_{x,y}$ can be the transition probabilities of the simple random walk on S.

Proposition 0.1 (Metropolis-Hastings dynamics)**.** *Let S be finite, and suppose we are given*

- *A **target measure** $m \in \mathcal{P}(S)$.*
- *A **proposal kernel** $Q = (q_{x,y})_{x,y}$, that is a Markov transition kernel.*
- *A **skew vorticity kernel** $R = (r_{x,y})_{x,y}$, such that $r_{x,y} = -r_{y,x}$ and $\sum_y r_{x,y} = 0$.*

Assume that $r_{x,y} \geq -m_y q_{y,x}$ for all $x, y \in S$ (in particular, since R is skew symmetric, $-m_y q_{y,x} \leq r_{x,y} \leq m_x q_{x,y}$). Then
$$p_{x,y} := \begin{cases} \min(q_{x,y}, \frac{r_{x,y}+m_y q_{y,x}}{m_x}) & \text{if } y \neq x \\ 1 - \sum_{z\neq x}\min(q_{x,z}, \frac{r_{x,z}+m_z q_{z,x}}{m_x}) & \text{if } y = x \end{cases} \tag{4}$$
*defines valid Markov transition probabilities, referred to as the (generalized) **Metropolis-Hastings kernel**, which admits m as invariant probability.*

Assume moreover that $q_{x,y}$ is irreducible and that $q_{x,y} > 0$ whenever $q_{y,x} > 0$. Then $p_{x,y}$ is irreducible and thus m is the unique invariant probability.

Proof. The normalization property $\sum_y p_{x,y} = 1$ just follows from the definition of $p_{x,x}$. We next check that $p_{x,y} \geq 0$. This is just the condition $r_{x,y} \geq -m_y q_{y,x}$ for the off-diagonal terms. While $p_{x,x} \geq 0$ since we can bound the sum with the negative sign by $\sum_{z\neq x} q_{x,z} = 1 - q_{x,x} \leq 1$.

We next verify that m is invariant. Indeed

$$\begin{aligned}\sum_{y\neq x} m_x p_{x,y} &= \sum_{y\neq x} \min(m_x q_{x,y}, r_{x,y} + m_y q_{y,x}) = \sum_{y\neq x} \min(m_x q_{x,y} - r_{x,y}, m_y q_{y,x}) + \sum_{y\neq x} r_{x,y} \\ &= \sum_{y\neq x} \min(m_x q_{x,y} + r_{y,x}, m_y q_{y,x}) = \sum_{y\neq x} m_y p_{y,x}\end{aligned}$$

where, from the first to the second line, we used $r_{x,y} = -r_{y,x}$ and $\sum_{y\neq x} r_{x,y} = 0$ (since $r_{x,x} = 0$). □

Metropolis-Hastings algorithms perform very well in several situations, however they feature some weak aspects:

- If our task is to converge to the invariant measure as fast as possible, having a chance $p_{x,x} > 0$ not to move at all, is hardly optimal.
- As β increases, the presence of local minima of U can increase dramatically the convergence time of the Markov chain.

Exercise 0.1. Check that, if $r = 0$, then m is reversible for the transition probabilities $p_{x,y}$ defined in Equation 4.

Exercise 0.2. Verify that hte Metropolis-Hastings dynamics include the aforementioned example: the trivial perfect simulation, the global heat bath, and the symmetric dynamics Equation 3. (Show that for a specific choice of $r_{x,y}$, $q_{x,y}$ one gets those dynamics).

Glauber and Kawasaki dynamics [Sketch]

Glauber dynamics is a specific instance of MCMC often used in statistical physics for spin systems (e.g., the Ising model), corresponding to a "single-site heat bath." In particular, we consider the case where S is a product space, say $S = \{-1, +1\}^V$, for some finite graph V. We interpret S as a *spin space*: V each point in V has a feature, say the spin, which can be either $+1$, or -1 at a given time. Moreover V is itself a graph, namely we have a notion of "neighbors" of a given $v \in V$. The target measure is the Gibbs measure $m(\sigma) \propto e^{-\beta H(\sigma)}$, where the Hamiltonian is a function $H\colon S \to \mathbb{R}$, e.g. $H(\sigma) = -\sum_{(u,v)\in E} J_{u,v}\sigma_u\sigma_v$. At each step, the dynamics proceeds as follows:

1. Select a vertex $v \in V$ uniformly at random.
2. Replace the spin σ_v with a new value $\sigma'_v \in \{-1, +1\}$ drawn from the conditional distribution of the spin at v given the configuration of its neighbors.

The transition probability to flip a spin at site v (changing σ to σ^v) is given by:

$$p_{\sigma,\sigma^v} = \frac{1}{|V|}\frac{e^{-\beta H(\sigma^v)}}{e^{-\beta H(\sigma)} + e^{-\beta H(\sigma^v)}} = \frac{1}{|V|}\frac{1}{1 + e^{\beta(H(\sigma^v) - H(\sigma))}}$$

This dynamics is reversible with respect to the Gibbs measure. A key feature is that the energy difference $H(\sigma^v) - H(\sigma)$ depends *only* on the neighbors of v, making the update computationally very cheap ($O(1)$ if the degree of the graph is bounded).

While Glauber dynamics allows the total "magnetization" (sum of spins) to change, **Kawasaki dynamics** is designed to sample from the Gibbs measure conditioned on a fixed number of $+1$ spins (canonical ensemble). For instance we can think in this case that $S = \{0, 1\}^V$, 0 representing an empty site, and 1 a site occupied by a particle. We want a dynamics that preserves the total number of particles. For instance:

1. Select a pair of vertices (u, v) connected by an edge in the graph.
2. Propose to **swap** the values of σ at u and v.

3. Accept the swap with a probability determined by the Metropolis rule:

$$p_{\sigma,\sigma^{u,v}} = \min\left(1, e^{-\beta(H(\sigma^{u,v})-H(\sigma))}\right)$$

where $\sigma^{u,v}$ is the configuration σ with the values at u and v swapped.

If $\sigma_u = \sigma_v$, the swap changes nothing and the energy is constant. If they differ, the particle moves. Since particles are only moved and never created or destroyed, the total number of $+1$ spins is conserved throughout the evolution of the chain.

Perfect Simulations

Finally we discuss a slightly different situation, where we want to sample the *unknown* invariant measure of a given Markov chain, so the transition probabilities are *given* in this case. We assume that the transition probabilities satisfy the following condition

$$Z := \sum_y \inf_{x\in S} p_{x,y} > 0 \tag{5}$$

In other words, there is at least one point on which the Markov chain can jump with uniformly positive probability, regardless of the initial point x. While this may seem very restrictive, even if $P = (p_{x,y})_{x,y}$ fails to satisfy Equation 5, maybe some power P^t will. And in this case, we can just apply the same technique to the t-multistep transition probability. In particular this will work for any aperiodic chain on a finite *or countable* state space.

Remark. Assuming Equation 5, we can write

$$p_{x,y} = (1-\varepsilon)q_{x,y} + \varepsilon\nu_y \tag{6}$$

for some $\varepsilon > 0$, $\nu \in \mathcal{P}(S)$ and Markov transition probabilities $Q = (q_{x,y})_{x,y}$. For instance we can take $\varepsilon = Z$ and

$$q_{x,y} = (p_{x,y} - \varepsilon\nu_y)/(1-\varepsilon)$$

Notice that, if $p_{x,y}$ is irreducible (or more generally considering only the communicating class containing the support of ν), then P induces a positive recurrent Markov chain (since at each step we have probability $\varepsilon\nu_y > 0$ to recur to some point y in the support of ν). In particular, the Markov chain admits a unique invariant distribution $m \in \mathcal{P}(S)$.

Proposition 0.2 (Perfect Simulation Algorithm)**.** *Assume an irreducible Markov chain on a finite or countable state space S, with transition probabilities $P = (p_{x,y})$, satisfies Equation 5, and let m be its invariant probability. For any decomposition $p_{x,y} = (1-\varepsilon)q_{x,y} + \varepsilon\nu_y$ as in Equation 6 the following holds.*

Let τ be a geometric random variable parameter ε, that is $\mathbb{P}(\tau = k) = \varepsilon(1-\varepsilon)^k$ for $k \in \mathbb{N}$. Let $\mathbf{Y} := (Y_t)$ be a Markov chain with transition probabilities $Q = (q_{x,y})$ and initial distribution ν. Then Y_τ has distribution m.

Proof. Let μ be the law of Y_τ, then for $y \in S$

$$\mu_y = \sum_{k=0}^{\infty} \mathbb{P}(Y_\tau = y | \tau = k)\mathbb{P}(\tau = k) = \sum_{k=0}^{\infty} \varepsilon(1-\varepsilon)^k \mathbb{P}(Y_k = y)$$

Therefore $\mu = \varepsilon\sum_k (1-\varepsilon)^k \nu Q^k$. We need to check $\mu = \mu P$, since we already know that the invariant measure is unique and the only solution to this equation is $\mu = m$.

In operator's notation $P = \varepsilon R + (1-\varepsilon)Q$, where $R_{x,y} = \nu_y$ for all $y \in S$. Notice that $\lambda R = \nu$ for all $\lambda \in \mathcal{P}(S)$. Therefore

$$\begin{aligned}\mu P &= \varepsilon \mu R + (1-\varepsilon)\mu Q = \varepsilon \nu + (1-\varepsilon)\left(\varepsilon \sum\nolimits_{k=0}^{\infty} (1-\varepsilon)^k \nu Q^k\right) Q = \varepsilon\left(\nu + \sum\nolimits_{k=0}^{\infty} (1-\varepsilon)^{k+1} \nu Q^{k+1}\right) \\ &= \varepsilon\left(\nu (1-\varepsilon)^0 \nu Q^0 + \sum\nolimits_{k=1}^{\infty} (1-\varepsilon)^k \nu Q^k\right) = \mu\end{aligned}$$

□

This means that, if ν and $q_{x,y}$ are computationally accessible, we have a very practical way to sample m:

1. Sample a geometric random variable τ with success parameter ε (e.g. count the number of failures till the first success when flipping a coin with ε).
2. Sample a point $x_0 \in S$ with distribution ν.
3. Sample τ steps of a Markov chain starting at x_0 with transition probabilities $(q_{x,y})$.
4. The position of the Markov chain after τ step samples m.

This approach is perfect, since we are not claiming any convergence in the long time asymptotics. It is just an exact sampling, although the number of iterations we need to simulate (Y_t) is random and (with small probability) possibly very long.