Cornell University, Fall 2021	CS 6820: Algorithms
Lecture notes: Markov chains	29 Nov – 6 Dec 2021

Markov chains model discrete-time random processes whose future state evolution depends only on the present state, not on the entire sequence of states leading up to the present. As such, they represent an important class of probabilistic models. However, in algorithm design they serve an important additional role: the most popular algorithmic procedure for sampling from complicated probability distributions is to design an appropriate Markov chain and simulate its state evolution. This method is known as *Markov Chain Monte Carlo (MCMC)*. In these notes we will present some aspects of the fundamental theory of Markov chains and of the MCMC paradigm for designing sampling algorithms.

Before delving into definitions, let us give some examples to illustrate what we mean by "sampling from complicated probability distributions."

Example 1. If G is a q-colorable graph then the uniform distribution on proper q-colorings of G is easy to define but potentially hard to sample. For example if $q \geq 3$ and G is allowed to be an arbitrary graph, it is NP-hard to decide if *any* q-coloring of G exists, let alone sample a uniformly random one.

Example 2. Generalizing the preceding example, given a graph G and two parameters β, γ, we may want to sample a random labeling of its vertices using labels in some set Σ, i.e. a random function $L : V(G) \to \Sigma$, with probability proportional to

$$\mu(L) = \prod_{(u,v)\in E(G)} \begin{cases} \beta & \text{if } L(u) = L(v) \\ \gamma & \text{if } L(u) \neq L(v). \end{cases}$$

The first example (sampling a random q-coloring) specializes this one by setting $|\Sigma| = q$, $\beta = 0$, $\gamma = 1$.

Example 3. Given a tuple of non-negative integers $(d_1, d_2, \ldots, d_n)$, consider the set of graphs with vertex set $[n] = \{1, 2, \ldots, n\}$ such that for all $i \in [n]$ the degree of vertex i is d_i. When this set is non-empty, one may wish to draw random samples from it. For example, sampling graphs from this distribution may be useful for simulating the performance of algorithms or distributed protocols on networks that resemble (in terms of their size and degree distribution) observed real-world network topologies. Alternatively, the ability to draw samples from this distribution may aid a statistician in testing the hypothesis that a network topology observed in the real world has some structure that is statistically distinguishable from random graphs with the same size and degree distribution.

Example 4. Suppose we are given:

1. a deep neural network (DNN) that generates random images by transforming an input layer of independent (Gaussian) random numbers into an output layer of pixels;

2. an image x with some missing pixels.

The DNN defines a probability distribution over output images (i.e., the distribution that results from feed-forward propagation of Gaussian random numbers at the input layer), and one may wish to draw samples from the conditional distribution over output images, conditioned on the pixel values matching the data present in x. For example, this sampling task may form part of the pipeline in an image completion algorithm: given a DNN that models natural scenes, and an image with a natural scene in the background and an object in the foreground that occludes part of the scene, the sampling algorithm could be used to generate hypothetical completions of the background image.

One can define the following class of algorithmic random sampling problems that includes all of the examples above, along with many other important and practical random sampling problems.

Definition 1. An *unnormalized distribution* on a finite set Ω is a function $\mu : \Omega \to \mathbb{R}_{\geq 0}$ such that

$$Z_\mu \stackrel{\Delta}{=} \sum_{\omega \in \Omega} \mu(\omega) > 0.$$

The corresponding probability distribution is $\bar{\mu}(\omega) = \mu(\omega)/Z_\mu$. Sampling from μ refers to the process of drawing a random sample $\omega \in \Omega$ with probability $\Pr(\omega) = \bar{\mu}(\omega)$. Approximately sampling from μ refers to any process that draws a random sample from Ω such that for all $\omega \in \Omega$,

$$(1 - \varepsilon)\bar{\mu}(\omega) \leq \Pr(\omega) \leq (1 + \varepsilon)\bar{\mu}(\omega)$$

for some specified approximation parameter ε.

One can often specify an unnormalized distribution μ by specifying an efficient algorithm to calculate $\mu(\omega)$ for every $\omega \in \Omega$. This brings us to the main question we address below.

> *Given an efficient algorithm for evaluating an unnormalized distribution $\mu(\omega)$, when is it possible to efficiently draw random samples from the probability distribution $\bar{\mu}$?*

Before continuing, let us pause to illustrate how the first and last examples above can be cast as special cases of this problem.

For the example of sampling a random q-coloring of a graph $G = (V, E)$, we can take Ω to be the set of all functions from V to $[q]$ (called "labelings" henceforth), and we can take μ to be a function that assigns the value 1 to labelings that are proper colorings of G and 0 to all other labelings. Then the probability distribution $\bar{\mu}$ is the uniform distribution on proper colorings of G.

For the example of image completion, we can take Ω to be the set of all functions that label each node of the DNN with a number called the node's *activation*.[1] We can then define

[1]Since our formalism requires Ω to be finite, we must quantize the set of numbers that can be used as a node's label. For example, we could limit the label set to be the set of 32-bit floating point numbers, or we could quantize node activations even more aggressively. Such quantization schemes have been advocated in the neural network literature, for the sake of making the training and inference process more efficient in terms of storage space, running time, and energy consumption.

$\mu(\omega)$ to be zero if the node activations in ω don't obey the DNN's weights and activation functions, or if the values in the output layer don't match the pixel values given in the input, x. However, when ω does obey the DNN's weights and activation functions and matches the given pixel values in the output layer, we define $\mu(\omega)$ to be the product of the (Gaussian) probabilities of the input node activations. Then the distribution $\bar{\mu}(\omega)$ is the conditional distribution defined in Example 4

1 Markov chains and their stationary distributions

In this section we formally define Markov chains, introduce the notion of a stationary distribution, and identify conditions under which a Markov chain has a unique stationary distribution such that the marginal distribution of the time-t state is guaranteed to converge to the stationary distribution as $t \to \infty$.

Definition 2. A Markov chain with (finite) state set Ω is a probability distribution on infinite sequences $X_0, X_1, \ldots$ of elements of Ω, satisfying the Markov property:

$$\forall t > 0 \, \forall (x_0, x_1, \ldots, x_t) \in \Omega^{t+1} \quad \Pr(X_t = x_t \mid X_0 = x_0, \ldots, X_{t-1} = x_{t-1}) = \Pr(X_t = x_t \mid X_{t-1} = x_{t-1}).$$

In other words, the conditional distribution of X_t depends only on the value of X_{t-1} and not on any of the values that came before time $t-1$.

A Markov chain is *time-homogeneous* if for all pairs $(x, y) \in S^2$, and all $t > 0$,

$$\Pr(X_t = x | X_{t-1} = y) = \Pr(X_{t+1} = x | X_t = y).$$

For a time-homogeneous Markov chain, the matrix P defined by $P_{xy} = \Pr(X_t = x | X_{t-1} = y)$ is called the *transition matrix*.

For the remainder of these lecture notes, all the Markov chains we consider will be time-homogeneous. Accordingly, when we use the term *Markov chain* below it always implicitly refers to a time-homogeneous Markov chain.

The probability distribution of a Markov chain's state at time t can be represented by a vector $\pi_t \in \mathbb{R}^\Omega$, whose x^{th} coordinate is the probability that $X_t = x$:

$$(\pi_t)_x = \Pr(X_t = x).$$

For $t > 0$ we can then calculate that

$$\begin{aligned}(\pi_t)_x = \Pr(X_t = x) &= \sum_{y \in \Omega} \Pr(X_t = x \wedge X_{t-1} = y) \\ &= \sum_{y \in \Omega} \Pr(X_t = x \mid X_{t-1} = y) \cdot \Pr(X_{t-1} = y) = \sum_{y \in \Omega} P_{xy} (\pi_{t-1})_y\end{aligned}$$

This can be summarized more succinctly as

$$\pi_t = P \pi_{t-1}$$

and, by induction, we obtain

$$\pi_t = P^t \pi_0.$$

Definition 3. A probability distribution π is a stationary distribution for a Markov chain with transition matrix P if it satisfies

$$P\pi = \pi.$$

A stationary distribution is thus a fixed point of the Markov chain's transition dynamics: if the initial state distribution π_0 is equal to the stationary distribution π, then every future state π_t is also distributed according to π.

It turns out that every Markov chain with finite state set has a stationary distribution. This fact, as well as a sufficient condition for the stationary distribution to be unique, can be deduced from the Perron-Frobenius Theorem, a fundamental theorem from linear algebra that concerns the eigenvalues of square matrices with non-negative entries.

Definition 4. If A is an $n \times n$ square matrix with non-negative entries, let G_A be the directed graph (potentially with self-loops) having vertex set $[n]$ and edge set $\{(i,j)|A_{ij} > 0\}$. We say A is *irreducible* if G_A is strongly connected, and we say A is *aperiodic* if the cycle lengths in G_A have no common divisor greater than 1.

Irreducible matrices are characterized by the property that every entry of $A + A^2 + A^3 + \ldots + A^n$ is strictly positive. Among irreducible matrices, the aperiodic ones are characterized by the property that for some positive integer k, every entry of A^k is strictly positive.

Theorem 1 (Perron-Frobenius)**.** *If A is an irreducible $n \times n$ square matrix with non-negative entries, then A has a unique right eigenvector $v \in \mathbb{R}^n$ whose components are strictly positive. The eigenvalue associated to v, called the* Perron-Frobenius eigenvalue*, has multiplicity one, and every other (complex) eigenvalue λ' satisfies $|\lambda'| \leq \lambda$. This inequality is strict if A is aperiodic.*

The proof of the Perron-Frobenius Theorem can be found in many linear algebra textbooks, for example Felix Gantmacher's *The Theory of Matrices* (AMS Chelsea Publishing, 2000). For the same of making these lecture notes self-contained, we will prove an easier result that pertains to Markov chain transition matrices.

Theorem 2. *If P is the transition matrix of an irreducible, aperiodic Markov chain with finite state set, then there is a unique stationary distribution π such that $P\pi = \pi$. For any starting distribution π_0, the time-t state distribution $\pi_t = P^t\pi_0$ converges to π as $t \to \infty$. In fact, the convergence is exponentially fast: there are constants $C < \infty$ and $\delta > 0$ such that*

$$\|\pi_t - \pi\|_1 \leq C(1-\delta)^t$$

for all $t \in \mathbb{N}$.

Proof. Since P is irreducible and aperiodic, there exists some k such that all entries of P^k are positive. Let $N = |\Omega|$ denote the number of states of the Markov chain, and choose $\varepsilon < 0$ such that all entries of P^k are greater than or equal to ε/N. Let $Q = (\mathbf{1}\mathbf{1}^\mathsf{T})/N$. Then

$$P^k = \varepsilon Q + (1-\varepsilon)R$$

where R is a non-negative matrix.

A *column-stochastic matrix* is a non-negative matrix whose column sums are all equal to 1. Equivalently, the non-negative matrix A is called column-stochastic if $\mathbf{1}^\mathsf{T} A = \mathbf{1}^\mathsf{T}$; from this characterization it is evident that the set of column-stochastic matrices is closed under multiplication. Note that Q is column-stochastic since $\mathbf{1}^\mathsf{T}\mathbf{1} = N$. Furthermore, P is column-stochastic since for every $y \in \Omega$ we have $\sum_x P_{xy} = \sum_{x\in\Omega} \Pr(X_t = x \mid X_{t-1} = y) = 1$. Hence P^k is column-stochastic, and we may conclude that R is also column-stochastic using the equation

$$(1-\varepsilon)\mathbf{1}^\mathsf{T} R = \mathbf{1}^\mathsf{T} P^k - \varepsilon \mathbf{1}^\mathsf{T} Q = \mathbf{1}^\mathsf{T} - \varepsilon \mathbf{1}^\mathsf{T} = (1-\varepsilon)\mathbf{1}^\mathsf{T}.$$

For $t \geq 0$ let $\Delta_t = \pi_{t+1} - \pi_t = (P^{t+1} - P^t)\pi_0$. We have

$$Q\Delta_t = \frac{1}{N}\mathbf{1}\mathbf{1}^\mathsf{T}(P^{t+1} - P_t)\pi_0 = 0,$$

since $\mathbf{1}^\mathsf{T}(P^{t+1} - P^t) = \mathbf{1}^\mathsf{T} - \mathbf{1}^\mathsf{T} = 0$. Therefore,

$$\Delta_{t+k} = P^k \Delta_t = (1-\varepsilon) R \Delta_t.$$

The inequality $\|Rv\|_1 \leq \|v\|_1$ holds for any vector v. To prove this, it suffices to verify it when $\|v\|_1 \leq 1$. A vector whose 1-norm is less than or equal to 1 is a convex combination of the standard basis vectors and their negations, hence we only need to check that $\|Rv\|_1 = 1$ when v is one of the standard basis vectors. In that case Rv is a column of R, i.e. a non-negative vector whose components sum up to 1, so $\|Rv\| = 1$. Now, using the inequality $\|Rv\|_1 \leq \|v\|_1$, we find that

$$\|\Delta_{t+k}\|_1 \leq (1-\varepsilon)\|\Delta_t\|.$$

For any $t \in \mathbb{N}$, if $q = \lfloor t/k \rfloor$, then

$$\begin{aligned}
\sum_{s=t}^{\infty} \|\Delta_s\| &\leq \sum_{r=q}^{\infty} \sum_{i=0}^{k-1} \|\Delta_{kr+i} \\
&\leq \sum_{r=q}^{\infty} \sum_{i=0}^{k-1} (1-\varepsilon)^r \|\Delta_i\| \\
&= \frac{(1-\varepsilon)^q}{\varepsilon} \left(\|\Delta_0\|_1 + \cdots + \|\Delta_{k-1}\|_1\right)
\end{aligned}$$

This confirms that the sequence $\pi_t = \pi_0 + \sum_{s=0}^{t-1} \Delta_s$ converges absolutely as $t \to \infty$ and that the rate of convergence is exponential. Denote the limit point by π. To conclude the proof we must show that π is a stationary distribution of P. The equation $P\pi = \pi$ follows by observing that

$$P\pi = \lim_{t\to\infty} (P\pi_t) = \lim_{t\to\infty} \pi_{t+1} = \pi.$$

The fact that π is a probability distribution follows from the fact that π_t is a probability distribution for each t, and that the set of probability distributions on $\mathbb{R}^\Omega$ is topologically closed. □

2 Reversible Markov chains and the Metropolis-Hastings algorithm

In general, computing the stationary distribution of a Markov chain requires solving a linear system, but there is one case in which the stationary distribution has a simple closed-form formula. This is the case of a reversible Markov chain.

Definition 5. A Markov chain with transition matrix P is reversible with respect to (unnormalized) distribution μ if it satisfies

$$P_{xy}\mu_y = P_{yx}\mu_x$$

for all $x, y \in \Omega$.

Lemma 3. *If P is reversible with respect to μ, then $\bar{\mu}$, the normalization of μ, is a stationary distribution for P.*

Proof. Multiplying both sides of the reversibility equation $P_{xy}\mu_y = P_{yx}\mu_x$ by the normalizing constant $\left(\sum_\omega \mu_\omega\right)^{-1}$, we find that $P_{xy}\bar{\mu}_y = P_{yx}\bar{\mu}_x$ for all $x, y \in \Omega$. Hence,

$$(P\bar{\mu})_x = \sum_{y \in S} P_{xy}\bar{\mu}_y = \sum_{y \in S} P_{yx}\bar{\mu}_x = \bar{\mu}_x.$$

□

The reversibility condition can be interpreted as a type of "detailed balance" condition: at stationarity, the rate of state transitions from x to y equals the rate of state transitions from y to x, for all state pairs x and y.

The Metropolis-Hastings algorithm is a procedure that takes an unnormalized distribution μ and creates a Markov chain P whose state transitions are computationally easy to simulate, and whose stationary distribution is $\bar{\mu}$. Actually the procedure makes use of an auxiliary Markov chain K, called the *proposal distribution*, whose stationary distribution is simple and often unrelated to μ. In many applications the stationary distribution of K is simply the uniform distribution on Ω. To define the Metropolis-Hastings algorithm we assume we have:

1. An unnormalized probability distribution specified by a function $\kappa : \Omega \to [0, 1]$.
2. A Markov chain K that is reversible with respect to κ.
3. Algorithms for sampling state transitions of K and for computing the function κ.

The Markov chain K is called the *proposal distribution* for the Metropolis-Hastings procedure. As stated earlier, in many applications $\bar{\kappa}$ is simply the uniform distribution over Ω, in which case $\kappa(\omega) = 1$ for all ω and the reversibility condition $K_{xy}\kappa_y = K_{yx}\kappa_x$ simply states that the Markov transition matrix K is a symmetric matrix.

Now for $x \neq y$ define

$$P_{xy} = K_{xy} \cdot \kappa_y \cdot \min\left\{\frac{\mu_x}{\mu_y}, 1\right\}, \tag{1}$$

and define $P_{yy} = 1 - \sum_{x \neq y} P_{xy}$. Note that

$$\sum_{x \neq y} P_{xy} = \left(\sum_{x \neq y} K_{xy} \cdot \min\left\{\frac{\mu_x}{\mu_y}, 1\right\}\right) \kappa_y \leq \left(\sum_{x \neq y} K_{xy}\right) \kappa_y \leq \kappa_y \leq 1,$$

so $P_{yy} \geq 0$. Thus, P is indeed a Markov transition matrix.

Lemma 4. *The Markov chain P defined by Equation (1) is reversible with respect to μ.*

Proof. Consider any $x, y \in \Omega$. If $x = y$ then the equation $P_{xy}\mu_y = P_{yx}\mu_x$ holds trivially. Otherwise,

$$P_{xy}\mu_y = K_{xy} \cdot \kappa_y \cdot \min\{\mu_x, \mu_y\}$$
$$P_{yx}\mu_x = K_{yx} \cdot \kappa_x \cdot \min\{\mu_y, \mu_x\}.$$

The lemma follows because $\min\{\mu_x, \mu_u\} = \min\{\mu_y, \mu_x\}$ and because our assumption that K is reversible with respect to κ implies $K_{xy}\kappa_y = K_{yx}\kappa_x$. □

An algorithm to simulate state transitions of the Markov chain P can be described as follows. Suppose the current state of the Markov chain is $y \in \Omega$.

1. Using the sampling oracle for Markov chain K, sample "proposed state" $x \in \Omega$ with probability K_{xy}.
2. Compute μ_x, μ_y, and κ_y.
3. With probability $\min\{\frac{\mu_x}{\mu_y}, 1\} \cdot \kappa_y$, transition to state x.
4. Otherwise, remain at state y.

Example 5 (Glauber dynamics for sampling q-colorings)**.** To illustrate the Metropolis-Hastings procedure, we show how to use it to define a simple Markov chain whose unique stationary distribution is the uniform distribution over proper q-colorings of a graph $G = (V, E)$. For two labelings $x, y : V \to [q]$ define their Hamming distance as

$$d(x, y) = \#\{v \in V \mid x(v) \neq y(v)\}.$$

Assume that q is large enough that the graph whose vertices are proper q-colorings of G, and whose edges are pairs of colorings whose Hamming distance is 1, constitutes a non-empty connected graph. (If this graph is not connected, the Markov chain defined here will be reducible and it will have multiple stationary distributions.)

We will take $\kappa(\omega) = 1$ for all $\omega \in \Omega$, and for our proposal distribution we will define $n = |V|$ and

$$K_{xy} = \begin{cases} \frac{1}{nq} & \text{if } d(x,y) = 1 \\ \frac{1}{q} & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases}$$

A state transition of K can be simulated by the following algorithm: starting from state y, sample vertex $v \in V$ and color $c \in [q]$ independently and uniformly at random, and let x be the state obtained from y by recoloring v with color c and leaving all other colors the same. From the definition of K it follows easily that $K_{xy} = K_{yx}$, i.e. K is reversible with respect to κ.

Recall that our goal is to design a Markov chain whose stationary distribution is the uniform distribution on proper q-colorings of G. In other words, we want to draw samples from the distribution given by the unnormalized density function μ such that $\mu(\omega) = 1$ when ω is a proper coloring and $\mu(\omega) = 0$ otherwise. To simulate a state transition of the Markov chain P defined by the Metropolis-Hastings procedure we do the following steps, starting from state y. Assume that y is a proper coloring.

1. *Sample "proposed state" $x \in \Omega$ with probability K_{xy}.*
   In other words, sample vertex $v \in V$ and color $c \in [q]$ independently and uniformly at random, and let x be the state obtained from y by recoloring v with color c and leaving all other colors the same.

2. *Compute μ_x, μ_y, and κ_y.*
   By assumption, y is a proper coloring, so $\mu_y = \kappa_y = 1$. Recall from above that $\mu_x = 1$ if and only if x is a proper coloring. Since y is a proper coloring and x is obtained from y by recoloring v, we only need to check whether every edge incident to v remains properly colored. In other words, to execute this step we merely need to test whether vertex v has any neighbor whose color is already c. If so, $\mu_x = 0$; otherwise, $\mu_x = 1$.

3. *With probability $\min\{\frac{\mu_x}{\mu_y}, 1\} \cdot \kappa_y$, transition to state x.*
   The probability in question is 1 if the color of every neighbor of v is different from c, and 0 otherwise.

4. *Otherwise, remain at state y.*

Hence, the Metropolis-Hastings Algorithm in this case corresponds to the following very simple procedure. The starting state of the Markov chain is any proper coloring of G. To simulate one state transition, we sample a uniformly random vertex v and uniformly random color c, and we change the color of v to c if and only if the color of every neighbor of v is different from c. This Markov chain on the set of proper colorings of G is called *Glauber dynamics*.