Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.437 Inference and Information
Spring 2015

21 Markov Chain Monte-Carlo in More Detail

In this section we revisit the Markov Chain Monte-Carlo (MCMC) methodology, developing the key ideas in more detail. We begin with a summary of some Markov chain background.

21.1 Markov Chain Summary

For these notes, it will be useful to keep in mind the following key results on steady-state distributions and mixing times of (finite-state) homogenous Markov chains (described in more detail in the appendix of these notes):

1. There may be multiple steady-state distributions. we find them by the solution to the (global) balance equations, which are the eigenvectors corresponding to the unit eigenvalues of the state transition matrix.

2. The rate at which we converge to the steady-state distribution is specified by the mixing time of the chain, which is governed by the eigenvalue having the second largest magnitude.

3. Markov chains that are reversible have the property that when operating in their steady-state distribution, the statistics of a sequence of states from the chain are unchanged if the sequence is reversed (i.e., whether time runs forwards or backwards).

4. The steady-state distributions of reversible markov chains can be found by solving the detailed (or local) balance equations, which can be simpler to solve than the global balance equations.

21.2 Monte Carlo Approximation

The generic inference problem of interest is to compute a good approximation to an expectation of the form

$$\gamma = \mathbb{E}\left[g(\mathsf{x})\right] = \sum_{x \in \mathcal{X}} g(x)\, p_{\mathsf{x}}(x) \tag{1}$$

for some function $g(\cdot)$ of interest, where $p_{\mathsf{x}}(\cdot)$ is the target distribution.

As our basic approach, we first generate i.i.d. samples $\mathsf{x}_1, \dots, \mathsf{x}_k$ from the distribution p_{x}, with some choice of k, then approximate (1) via

$$\hat{\gamma} \cong \frac{1}{k} \sum_{j=1}^{k} g(\mathsf{x}_j). \tag{2}$$

This is called a Monte Carlo approximation.

In general, in an appropriate sense we will develop in a forthcoming lecture, $\hat{\gamma} \to \gamma$ as $k \to \infty$. So we simply have to pick k large enough. Now of course for (2) to be computationally more efficient than (1), we need $k \ll |\mathcal{X}|$ to suffice for a good approximation. Fortunately, this is often the case. Indeed, with many models, $p_{\mathsf{x}}(x)$ is negligibly small for many values of $x \in \mathcal{X}$. Thus, the summation (1) involves many terms in the sum that are effectively zero; in essence this is an expression of the structure in the distribution. With Monte Carlo approximation we effectively avoid including the terms in (1) that are negligibly small. In particular, by the use of sampling, we are only going to obtain samples from p_{x} that have significant probability—the other values simply won't be generated. Thus, we only need k to be big enough that we end up generating all the values x that have significant probability.

21.3 Rejection Sampling

In principle, the preceding development can be used to generate samples regardless of the alphabet size $m = |\mathcal{X}|$. However, in practice, when m is large we need to accommodate the fact that we generally know the target distribution p_{x} only to within a constant of proportionality, due to the difficulty of computing the partition function. In particular, we generally have knowledge of $\tilde{p}_{\mathsf{x}}(x)$ where

$$p_{\mathsf{x}}(x) = \frac{1}{Z}\,\tilde{p}_{\mathsf{x}}(x).$$

In such cases, one conceptually important approach to simulation is referred to as the *rejection method.* To apply this method, we assume we have a distribution q_{x} on $\mathcal{X}$ that we know we can sample from. We call q_{x} the *proposal distribution.* For example, q_{x} might be the uniform distribution $\mathtt{U}(\mathcal{X})$. In addition, we assume we know how to efficiently find a constant c such that

$$\tilde{p}_{\mathsf{x}}(x) \le c\, q_{\mathsf{x}}(x), \qquad \text{all } x \in \mathcal{X}. \tag{3}$$

With rejection sampling, we start by generating samples from the proposal distribution q_{x}. It is convenient to visualize a collection of m bins, one associated with each value $x \in \mathcal{X}$, so this process puts some number of samples in each bin, in proportion to the "shape" of the distribution q_{x}. Then we "re-shape" the resulting distribution according to $\tilde{p}_{\mathsf{x}}$ by randomly discarding a fraction of the generated samples from each bin. The constraint (3) effectively ensures such re-shaping is possible.

A formal description of the procedure is as follows.

1. Generate a random sample x' from the distribution $q_{\mathsf{x}}(\cdot)$.

2. Given the sample x', generate a random sample u from a Bernoulli distribution p_{u} on $\mathcal{U} = \{\mathrm{A}, \mathrm{R}\}$ with

$$p_{\mathsf{u}|\mathsf{x}'}(\mathrm{A}|x') = \frac{\tilde{p}_{\mathsf{x}}(x')}{c\, q_{\mathsf{x}}(x')}.$$

3. Let
$$y \triangleq \begin{cases} x' & u = \mathrm{A} \\ \mathrm{E} & u = \mathrm{R}, \end{cases}$$
where E denotes the "erasure" symbol.

4. If $y \neq \mathrm{E}$, let $x = y$, i.e., keep the sample. Otherwise discard the generated sample.

This algorithm is executed repeatedly (and independently) until the desired number (k) of samples from p_{x} are accumulated. Let k' denote the required number of iterations of the algorithm.

Let's verify that the samples produced have the desired distribution, then examine the sample acceptance rate.

First, we show that an accepted sample has the correct distribution by the following calculation:

$$\begin{aligned}
\mathbb{P}(\mathsf{x} = x) &= \mathbb{P}(\mathsf{y} = x \mid \mathsf{u} = \mathrm{A}) \\
&= \mathbb{P}(\mathsf{x}' = x \mid \mathsf{u} = \mathrm{A}) \\
&= p_{\mathsf{x}'|\mathsf{u}}(x|\mathrm{A}) \\
&= \frac{p_{\mathsf{u}|\mathsf{x}'}(\mathrm{A}|x)\, q_{\mathsf{x}}(x)}{\sum_{a\in\mathcal{X}} p_{\mathsf{u}|\mathsf{x}'}(\mathrm{A}|a)\, q_{\mathsf{x}}(a)} \\
&= \frac{\left(\frac{\tilde{p}_{\mathsf{x}}(x)}{c\, q_{\mathsf{x}}(x)}\right) q_{\mathsf{x}}(x)}{\sum_{a\in\mathcal{X}} \left(\frac{\tilde{p}_{\mathsf{x}}(a)}{c\, q_{\mathsf{x}}(a)}\right) q_{\mathsf{x}}(a)} \\
&= \frac{\tilde{p}_{\mathsf{x}}(x)}{\sum_{a\in\mathcal{X}} \tilde{p}_{\mathsf{x}}(a)} \\
&= \frac{\tilde{p}_{\mathsf{x}}(x)}{Z} \\
&= p_{\mathsf{x}}(x),
\end{aligned}$$

as desired.

Next, we calculate the fraction (k/k') of samples accepted via

$$\begin{aligned}\mathbb{P}(\mathsf{y} \neq \mathrm{E}) &= \mathbb{P}(\mathsf{u} = \mathrm{A}) \\ &= p_{\mathsf{u}}(\mathrm{A}) \\ &= \sum_{a\in\mathcal{X}} p_{\mathsf{u}|\mathsf{x}'}(\mathrm{A}|a)\, q_{\mathsf{x}}(a) \\ &= \sum_{a\in\mathcal{X}} \left(\frac{\tilde{p}_{\mathsf{x}}(a)}{c\, q_{\mathsf{x}}(a)}\right) q_{\mathsf{x}}(a) \\ &= \frac{1}{c}\sum_{a\in\mathcal{X}} \tilde{p}_{\mathsf{x}}(a) \\ &= \frac{Z}{c}.\end{aligned}$$

Hence, the smaller c is—while still meeting the constraint (3)—the more efficient the algorithm is (in terms of requiring the smallest k' to obtain k accepted samples.

Example 1 (Rejection Sampling from Binomial Distribution)**.** As an illustration of rejection sampling, suppose we want to use the method to generate samples from the symmetric binomial distribution

$$p_{\mathsf{x}}(x) = \binom{m}{x}\frac{1}{2^m}, \qquad x \in \mathcal{X} = \{0, 1, \dots, m\},$$

where, for convenience, m is even. Since we have a normalized version of the target distribution, $Z = 1$. Let us choose the proposal distribution to be uniform, i.e., $q_{\mathsf{x}} = \mathtt{U}(\mathcal{X})$. To determine a suitable constant c in (3), note that $p_{\mathsf{x}}(x)$ achieves its maximum at $x = m/2$. Moreover, using, for example, Stirling's approximation $n! \cong n^n e^{-n}$, it follows that for large m this probability is close to one, i.e.,

$$p_{\mathsf{x}}(x) \leq p_{\mathsf{x}}(m/2) = \binom{m}{m/2}\frac{1}{2^m} \cong 2^m.\frac{1}{2^m} = 1.$$

As a result, the minimum c we can accommodate in (3) for large m is $c = m + 1$, since $q_{\mathsf{x}}(x) = 1/(m+1)$, all x. Thus, in this regime, the sample acceptance rate is

$$\frac{Z}{c} = \frac{1}{m+1}$$

which approaches zero as $m \to \infty$. Hence, rejection sampling is quite inefficient when the alphabet size is large.

The behavior observed in Example 1 is representative: when sampling from distributions with large alphabets, it is generally quite inefficient. Indeed, it is generally prohibitively so.

This behavior makes rejection sampling impractical in such scenarios. However, as we will develop in the next installment of the notes, generalizations of the rejection sampling concept are extraordinarily practical.

21.4 Metropolis-Hastings Algorithm

While rejection sampling does not directly lead to a practical method for generating samples from a distribution over a large alphabet, a basic modification of rejection sampling does. In particular, if instead of requiring rejection sampling to produce independent samples, we allow it to produce suitably *dependent* samples, a practical algorithm is possible.[1] We do this by making the target distribution p_{x} the steady-state distribution of a Markov chain, and let the samples $\mathsf{x}_1, \mathsf{x}_2, \ldots$ be the samples from a realization (sample sequence) of the chain. This is the essence of MCMC.

We now develop the key principles and perspectives.

We want to construct a homogeneous Markov chain with the steady-state distribution p_{x}. The alphabet $\mathcal{X}$ is large enough that we assume it is computationally infeasible to normalize the distribution p_{x}, so what we know is $\tilde{p}_{\mathsf{x}}(x)$ where

$$p_{\mathsf{x}}(x) = \frac{1}{Z}\,\tilde{p}_{\mathsf{x}}(x).$$

Furthermore, we assume there is some homogeneous Markov chain we can implement. Let $V(x'|x)$ denote its defining transition distribution. We call $V(x'|x)$ a *proposal* transition distribution.

The following computationally very simple procedure transforms this Markov chain into a new reversible Markov chain with p_{x} as its unique steady-state distribution.[2]

1. Start with an arbitrary initial state x_0.

2. At time n, suppose a sample $\mathsf{x}_n = x$ has been generated.

3. Generate a proposed new state x' as a sample of the random variable x' distributed according to the proposal distribution $V(\cdot|x)$.

4. Compute acceptance factor

$$a(x \to x') \triangleq \min\left\{1\,,\, \frac{\tilde{p}_{\mathsf{x}}(x')\,V(x|x')}{\tilde{p}_{\mathsf{x}}(x)\,V(x'|x)}\right\}. \tag{4}$$

5. Generate a sample u from a Bernoulli random variable $\mathsf{u} \in \mathcal{U} = \{\mathrm{A}, \mathrm{R}\}$ with

$$\mathbb{P}(\mathsf{u} = \mathrm{A}|\mathsf{x}_n = x, \mathsf{x}' = x') = a(x \to x').$$

[1] While beyond the scope of our current treatment, even though the samples are dependent, the approximation (2) will still converge to the desired expectation with enough samples. However, the rate of convergence will be determined by the number of effectively independent samples in the sample average.

[2] Ideally, we'd also like the chain to have a short mixing time so that we converge quickly to the steady-state, but this is harder to ensure.

6. If $u = \mathrm{A}$, accept the new state by setting $\mathsf{x}_{n+1} = x'$. Otherwise, reject the new state and keep $\mathsf{x}_{n+1} = x$,

7. Iterate from Step 2.

This procedure is referred to as the Metropolis-Hasting (MH) algorithm for MCMC, and was originally developed for solving problems of statistical physics, but has since become extraordinarily widely used in inference more generally.

Note that the first several samples obtained from the algorithm are discarded, since it takes time before the chain converges to its steady-state distribution. The number of discarded samples is thus determined by the mixing time of the chain, which must be estimated (or guessed). This is referred to as the "burn-in" period for the algorithm.

21.5 How Metropolis-Hasting Works

The MH algorithm transforms the transition distribution $V(\cdot\,|\,\cdot)$ into a new transition distribution $W(\cdot\,|\,\cdot)$ that satifies the detailed balance equations with respect to our desired steady-state distribution p_{x}, i.e.,

$$p_{\mathsf{x}}(x)\,W(x'|x) = p_{\mathsf{x}}(x')\,W(x|x'), \quad \text{all } x, x' \in \mathcal{X}. \tag{5}$$

Note first that (5) is automatically satisfied by any choice of $W(\cdot\,|\,\cdot)$ when $x = x'$. Hence, we need only concern ourselves with the case $x \neq x'$.

Note, too, that there are many different transition distributions W that satisfy the detailed detailed balance equations (5). For example, if $W(\cdot\,|\,\cdot)$ is a solution to (5), so is $\tilde{W}(\cdot\,|\,\cdot)$ where

$$\tilde{W}(x'|x) = \alpha\,W(x'|x), \qquad \text{for all } x \neq x'. \tag{6}$$

for any $\alpha < 1$.[3]

Next, note that the effective transition matrix of the chain created by MH is, for all $x' \neq x$,

$$\begin{aligned} W(x'|x) &= \mathbb{P}\left(\mathsf{x}_{n+1} = x'|\mathsf{x}_n = x\right) \\ &= \mathbb{P}\left(\mathsf{x}'_{n+1} = x', \mathsf{u} = \mathrm{A}|\mathsf{x}_n = x\right) & (7) \\ &= \mathbb{P}\left(\mathsf{x}'_{n+1} = x'|\mathsf{x}_n = x\right)\,\mathbb{P}\left(\mathsf{u} = \mathrm{A}|\mathsf{x}'_{n+1} = x', \mathsf{x}_n = x\right) & (8) \\ &= V(x'|x)\,a(x \to x'), & (9) \end{aligned}$$

where (7) follows from the fact that the only way that $x' \neq x$ can arise is if the new state is accepted, and where (8) is an application of the chain rule of probability.

[3] Evidently, in this case $\tilde{W}(x|x) > W(x|x)$ to ensure $\tilde{W}$ is a valid distribution.

Now let's see how MH imposes detailed balance. Suppose for some states x and x' that the given chain specified by V is not balanced with respect to our desired steady-state distribution p_{x}. In particular, without loss of generality, suppose that for these states we have the imbalance

$$p_{\mathsf{x}}(x)\, V(x'|x) > p_{\mathsf{x}}(x')\, V(x|x').$$

Then choosing the acceptance factors

$$\begin{aligned} a(x \to x') &= \frac{\tilde{p}_{\mathsf{x}}(x')\, V(x|x')}{\tilde{p}_{\mathsf{x}}(x)\, V(x'|x)} = \frac{p_{\mathsf{x}}(x')\, V(x|x')}{p_{\mathsf{x}}(x)\, V(x'|x)} \\ a(x' \to x) &= 1 \end{aligned}$$

corrects the imbalance, i.e.,

$$\begin{aligned} p_{\mathsf{x}}(x)\, W(x'|x) &= p_{\mathsf{x}}(x)\, V(x'|x)\, a(x \to x') \\ &= p_{\mathsf{x}}(x)\, V(x'|x)\, \frac{p_{\mathsf{x}}(x')\, V(x|x')}{p_{\mathsf{x}}(x)\, V(x'|x)} \\ &= p_{\mathsf{x}}(x')\, V(x|x') \\ &= p_{\mathsf{x}}(x')\, V(x|x')\, a(x' \to x) \\ &= p_{\mathsf{x}}(x')\, W(x|x'), \end{aligned} \tag{10}$$

and thus the effective chain created by MH has p_{x} as a steady-state distribution.

Note that it is the requirement that the induced chain W be reversible that allows us to avoid having to know the partition function (normalization) Z for the distribution p_{x}. Indeed, in contrast to the global balance equations, the local nature of detailed balance (5) means that such normalizations are not needed.

Note, too, that our choice of acceptance factors was not unique. For example, if we instead use the reduced acceptance factors

$$\begin{aligned} \tilde{a}(x \to x') &= \alpha_{x,x'}\, a(x \to x') \\ \tilde{a}(x' \to x) &= \alpha_{x,x'}\, a(x' \to x), \end{aligned} \tag{11}$$

where $\alpha_{x,x'} < 1$ is a constant, then detailed balance will continue to be satisfied, as is easily verified by repeating the steps leading to (10). However, this means that the acceptance rate will be lower, which means that the chain will reject more samples and evolve more slowly. This, in turn, implies that the rate of decay of dependence among variables in the chain is slow, which means the chain must be run a lot longer to obtain a given number of independent samples.

A Rejection Sampling Perspective on MH

Consider specializing MH to the case when the proposal transition distribution corresponds to the special case of a Markov chain in which the samples are i.i.d., i.e.,

$V_0(x'|x) = q_{\mathsf{x}}(x')$ for all x, x', the transition probability does not depend on the current state x.

From our development of rejection sampling, we know that to obtain i.i.d. samples from p_{x}, we accept a sample x' generated from $V_0(\cdot|x) = q_{\mathsf{x}}(\cdot)$ with a probability

$$\mathbb{P}(\mathsf{u} = \mathrm{A}|\mathsf{x}_n = x, \mathsf{x}' = x') = a \triangleq \frac{p_{\mathsf{x}}(x')}{c\, q_{\mathsf{x}}(x')} \tag{12}$$

for some c chosen large enough that

$$p_{\mathsf{x}}(x') \leq c\, q_{\mathsf{x}}(x'), \quad \text{for all } x'. \tag{13}$$

Note that requirement on c is a *global* one, which is the root of the inefficiency of conventional rejection sampling.

From this perspective, MH can be viewed as a local version of rejection sampling, which allows more flexibility in the choice of c. To see this, consider replacing $q_{\mathsf{x}}(x')$ with a more general proposal distribution $V(x'|x)$ and applying MH with the global rejection sampling acceptance factor

$$\mathbb{P}(\mathsf{u} = \mathrm{A}|\mathsf{x}_n = x, \mathsf{x}' = x') = a \triangleq \frac{p_{\mathsf{x}}(x')}{c\, V(x'|x)} \tag{14}$$

instead of the usual MH one. Then it follows immediately that detailed balance (with respect to p_{x}) is satisfied with this acceptance factor:

$$\begin{aligned} p_{\mathsf{x}}(x)\, W(x'|x) &= p_{\mathsf{x}}(x)\, V(x'|x)\, a \\ &= p_{\mathsf{x}}(x)\, V(x'|x) \left(\frac{p_{\mathsf{x}}(x')}{c\, V(x'|x)} \right) \\ &= \frac{1}{c}\, p_{\mathsf{x}}(x)\, p_{\mathsf{x}}(x') \\ &= p_{\mathsf{x}}(x')\, W(x|x'), \end{aligned}$$

where the last equality follows from symmetry.

Hence, this chain will produce samples from the desired distribution, but the rate at which effectively independent samples are produced is very low because c must be large enough to meet its global constraint (13).

From this perspective, the key insight of MH is that it is not necessary to choose the acceptance rate (via c) to satisfy the global constraint (13); it need only satisfy a local one corresponding to detailed balance (5). And because the constraint is local, c can depend on x and x', as the discussion surrounding (11) established. Accordingly, replacing c with $c_{\mathsf{x},\mathsf{x}'}$ in (14) we obtain the acceptance rate

$$\mathbb{P}(\mathsf{u} = \mathrm{A}|\mathsf{x}_n = x, \mathsf{x}' = x') = a_c(x \to x') \triangleq \frac{p_{\mathsf{x}}(x')}{c_{x,x'}\, V(x'|x)}. \tag{15}$$

Since smaller choices for $c_{x,x'}$ mean a higher acceptance rate and thus more efficient sample generation, we want to choose these factors as small as possible subject to the constraint that (15) is a valid probability (i.e., takes a value that is at most 1). Thus, we reduce $c_{x,x'}$ until

$$\max\left\{a_c(x \to x'), a_c(x' \to x)\right\} = \max\left\{\frac{p_{\mathsf{x}}(x')}{c_{x,x'}\, V(x'|x)}, \frac{p_{\mathsf{x}}(x)}{c_{x,x'}\, V(x|x')}\right\} = 1$$

i.e., until one of the acceptance factor is saturated to value 1. This happens precisely when

$$c_{x,x'} = \max\{\frac{p_{\mathsf{x}}(x')}{V(x'|x)}, \frac{p_{\mathsf{x}'}(x)}{V(x|x')}\}, \tag{16}$$

corresponding to the MH acceptance factor values given by (4), i.e.,

$$a_c(x \to x') = a(x \to x').$$

Choosing a proposal distribution

An attractive aspect of the MCMC in general, and MH in particular, is the flexibility in the choice of proposal distribution $V(\cdot\,|\,\cdot)$, which need have no connection to the distribution p_{x} from which we want samples. What is important is that the Markov chain associated with the proposal transition distribution be such that all states in $\mathcal{X}$ are "visited" arbitarily often by the chain over time. Equivalently, we require that the steady-state distribution q_{x} associated with the Markov chain with transition distribution given by $V(\cdot\,|\,\cdot)$ be unique and $q_{\mathsf{x}}(x) > 0$, all $x \in \mathcal{X}$.

This is generally easy to do. It is sufficient, for example, to impose that $V(x'|x) > 0$ for all $x', x \in \mathcal{X}$. However, in practice such an overly stringent constraint is undesirable—for large alphabets it can be hard to sample from such a $V(\cdot|x)$. As a special case, one could even choose the proposal distribution to not depend on the current state at all, i.e., $V(\cdot|x) = q_{\mathsf{x}}(\cdot)$ provided $q_{\mathsf{x}}(x') > 0$ for all $x' \in \mathcal{X}$, which suffers from the same problem. The advantage of allowing a $V(\cdot|x)$ that depends on x is that there is no need for it to sample from all of $\mathcal{X}$. As an example, suppose $\mathcal{X} = \{0, 1, \ldots, m-1\}$ where m is large. Then instead of trying to sample from a distribution q_{x} over this large alphabet, we can sample from

$$V(x'|x) = \begin{cases} 1/2 & x' = (x+1) \bmod m \\ 1/2 & x' = (x-1) \bmod m, \end{cases} \tag{17}$$

which for each x is over a small (binary) alphabet. This proposal distribution can be viewed as taking a random talk over the alphabet of states $\mathcal{X}$. Eq. (17) is an example of a symmetric proposal distribution, i.e., one for which $V(x'|x) = V(x|x')$.[4] The symmetry constraint has the added advantage that the acceptance factor $a(x \to x')$ no longer depends explicitly on $V(\cdot\,|\cdot)$.

[4] When, as required, the steady-state distribution associated with such a symmetric V is unique, the corresponding q_{x} is uniform and the Markov chain associated with V happens to be reversible.

A Markov Chains: Steady-State and Mixing Time

Here we summarize some relevant aspects of the behavior of sequences of random variables x_1^n that form a *Markov chain*

$$\mathsf{x}_1 \leftrightarrow \mathsf{x}_2 \leftrightarrow \cdots \leftrightarrow \mathsf{x}_n.$$

A.1 Model and Questions

We focus on *finite-state, homogeneous* Markov chains. In particular, the variables $\mathsf{x}_1, \mathsf{x}_2, \ldots$ all take values in a common (finite) alphabet $\mathcal{X}$ of size $m = |\mathcal{X}| < \infty$, and their joint relationship is governed by a common state transition distribution $W(\cdot \mid \cdot)$, i.e.,

$$p_{\mathsf{x}_{n+1}|\mathsf{x}_n}(a|a') \triangleq W(a|a'), \qquad n = 1, 2, \ldots$$

Hence, given an initial distribution $p_{\mathsf{x}_1}(\cdot)$, the joint distribution for the finite set of variables x_1^n is

$$p_{\mathsf{x}_1^n}(x_1^n) = p_{\mathsf{x}_1}(x_1) \prod_{i=1}^{n-1} W(x_{i+1}|x_i). \tag{18}$$

When we view this chain as continuing indefinitely, the marginal distributions

$$p_{\mathsf{x}_1}(\cdot), p_{\mathsf{x}_2}(\cdot), p_{\mathsf{x}_3}(\cdot), \ldots$$

evolve according to the iteration

$$p_{\mathsf{x}_{n+1}}(x_{n+1}) = \sum_{x_n \in \mathcal{X}} p_{\mathsf{x}_n}(x_n)\, W(x_{n+1}|x_n), \qquad n = 1, 2, \ldots \tag{19}$$

For such chains, there exists a distribution to which this sequence of marginals will converge according to this iteration. We will summarize: 1) how to find such distribution; 2) the progression to it; and 3) the rate at which we approach it.

Example 2 (Two-State Markov Chain)**.** The following simplified two-state Markov chain will be useful in illustrating some aspects of the behavior of such problems. Suppose we study the performance of a particular stock in the market. We note that some days the stock is "up" on the day, and "down" on others, and that the up and down days come in streaks. A model for capturing such behavior is the two-state Markov chain whose state transition diagram is depicted in Fig. 1, where U denotes the "up" state, and D the "down" state. If the stock is up today, the probability that it will be down tomorrow is $0 < \alpha < 1$, whereas if it is down today, the probability that it will be up tomorrow is $0 < \beta < 1$. Hence, if α and β are both smaller than $1/2$, then this model will give rise to up and down days coming much more in bunches than than would be the case if the up and down days were independent and equally likely (i.e., $\alpha = \beta = 1/2$). Suppose someone has identified suitable values for α and

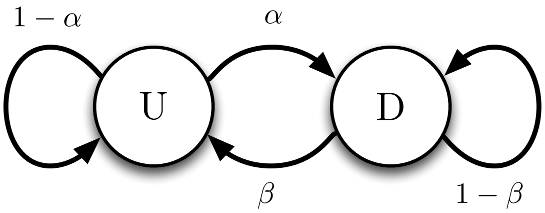

Figure 1: The state transition diagram for a two-state Markov chain.

β for a stock of interest. Among other things we might want to know are: for what values of α and β will the stock be a good acquisition over the long run, and in such cases how quickly can the investor expect to be rewarded?

Ultimately, we are concerned with how the marginal distributions in a Markov chain evolves, what *steady-state* it reaches, and how it reaches it, which we now develop.

A.2 Steady-State

As we have discussed, the marginal distributions of a finite-state, homogenous Markov chain converge over time. Specifically,

$$p_{\mathsf{x}_n}(a) \to p_{\mathsf{x}}(a) \quad \text{as} \quad n \to \infty, \quad \text{for all } a \in \mathcal{X},$$

where p_{x} is the *steady-state* distribution to which this sequence of distributions is evolving. From (19), we see that p_{x} must be a solution to the following equations for the steady-state:

$$p_{\mathsf{x}}(a) = \sum_{a' \in \mathcal{X}} p_{\mathsf{x}}(a')\, W(a|a'), \quad \text{for all } a \in \mathcal{X}. \tag{20}$$

Eqs. (20) are referred to as the *(global) balance equations*.

As equivalent terminology, we say that the sequences of variables x_n *converges in distribution* to x, where x is distributed according to the steady-state distribution p_{x}, which we write as

$$\mathsf{x}_n \xrightarrow{\mathrm{d}} \mathsf{x}, \quad \text{as } n \to \infty. \tag{21}$$

In particular, (21) says that the $x_1, x_2, \ldots$ are approaching some realization x of x. Rather, it is simply saying their *distributions* $p_{\mathsf{x}_1}, p_{\mathsf{x}_2}, \ldots$ are approaching p_{x}.

Example 3 (Two-State Markov Chain, Revisited)**.** Let's figure out the steady-state distribution of the two-state Markov Chain depicted in Fig. 1. Since there are two-states, i.e., $\mathcal{X} = \{\mathrm{U}, \mathrm{D}\}$, (20) consists of the following two equations:

$$\begin{aligned} p_{\mathsf{x}}(\mathrm{U}) &= p_{\mathsf{x}}(\mathrm{U}) W(\mathrm{U}|\mathrm{U}) + p_{\mathsf{x}}(\mathrm{D}) W(\mathrm{U}|\mathrm{D}) \\ &= p_{\mathsf{x}}(\mathrm{U})\,(1-\alpha) + p_{\mathsf{x}}(\mathrm{D})\,\beta \qquad (22) \\ p_{\mathsf{x}}(\mathrm{D}) &= p_{\mathsf{x}}(\mathrm{U}) W(\mathrm{D}|\mathrm{U}) + p_{\mathsf{x}}(\mathrm{D}) W(\mathrm{D}|\mathrm{D}) \\ &= p_{\mathsf{x}}(\mathrm{U})\,\alpha + p_{\mathsf{x}}(\mathrm{D})\,(1-\beta) \qquad (23) \end{aligned}$$

Rearranging each of (22) and (23), we note they express the same relationship, viz.,

$$\alpha\, p_{\mathsf{x}}(\mathrm{U}) = \beta\, p_{\mathsf{x}}(\mathrm{D}). \qquad (24a)$$

Clearly we need one more equation to be able to solve for $p_{\mathsf{x}}(\mathrm{U})$ and $p_{\mathsf{x}}(\mathrm{D})$, which is the normalization equation, i.e.,

$$p_{\mathsf{x}}(\mathrm{U}) + p_{\mathsf{x}}(\mathrm{D}) = 1. \qquad (24b)$$

Solving the simultaneous equations (24), we obtain

$$p_{\mathsf{x}}(\mathrm{U}) = \frac{\beta}{\alpha+\beta} \quad \text{and} \quad p_{\mathsf{x}}(\mathrm{D}) = \frac{\alpha}{\alpha+\beta}. \qquad (25)$$

Hence, over the long run, the stock is up a fraction $\beta/(\alpha+\beta)$ of the time. Evidently, if $\beta > \alpha$, this stock would be a good purchase over the long run, as the stock would be up more than half the time.

The kind of behavior observed in Example 3 is always what we experience in solving the steady-state equations, even for larger alphabet sizes. In particular, there is always redundancy among the equations, i.e., there is always at least one equation that is a consequence of the others, and to solve for the steady-state distribution values we need to impose the normalization constraint.

Now that we've learned how to calculate steady-state distributions, let's discuss what they are. By definition, the steady-state distribution is such that if the chain starts with that distribution, it remains in that distribution with each time step. All Markov chains have such a distribution. However, they may have several—for example, if

$$W(i|j) = \mathbb{1}(i=j)$$

then any distribution is a steady-state distribution! In such cases, all we can say is that depending on the initial state distribution, the chain will converge to one of the steady-state distributions. It is possible to develop necessary and sufficient conditions on the transition distribution for the steady-state distribution to be unique, though such a development is beyond our scope. However, a simple sufficient condition for uniqueness is that $W(i|j) > 0$ for all i, j.

A.3 Mixing Time

Now that we know the distribution of variables in a Markov chain converge to a steady-state distribution, we'd like to know the number of time steps it takes them to converge, and more generally exactly how they behave as they are converging.

To analyze such characteristics, it is helpful to use a little linear algebra. Accordingly, let's adopt the following matrix-vector nation: with, again, $m = |\mathcal{X}|$, we let

$$\mathbf{W} = \begin{bmatrix} W(1|1) & W(2|1) & \cdots & W(m|1) \\ W(1|2) & W(2|2) & \cdots & W(m|2) \\ \vdots & \vdots & \ddots & \vdots \\ W(1|m) & W(2|m) & \cdots & W(m|m) \end{bmatrix},$$

denote the state transition matrix, and

$$\mathbf{p}_{\mathsf{x}} = \begin{bmatrix} p_{\mathsf{x}}(1) & p_{\mathsf{x}}(2) & \cdots & p_{\mathsf{x}}(m) \end{bmatrix}.$$

denote the steady-state distribution (row) vector. Hence, $[\mathbf{W}]_{i,j} = w_{ij} = W(j|i)$ and $[\mathbf{p}_{\mathsf{x}}]_i = p_i = p_{\mathsf{x}}(i)$. Note that the matrix $\mathbf{W}$ has some special properties: its entries are nonnegative, and the rows sum to one. Such a matrix is referred to as a *stochastic* matrix.[5]

With this notation, the steady-state distributions are the solutions to the vector equation

$$\mathbf{p}_{\mathsf{x}} = \mathbf{p}_{\mathsf{x}}\, \mathbf{W}.$$

More generally, with $\mathbf{p}_{\mathsf{x}_n}$ denoting the vector form of the marginal distribution at time n, we have

$$\mathbf{p}_{\mathsf{x}_{n+1}} = \mathbf{p}_{\mathsf{x}_n} \mathbf{W}$$

and, thus,

$$\mathbf{p}_{\mathsf{x}_{n+1}} = \mathbf{p}_{\mathsf{x}_1} \mathbf{W}^n$$

Hence to understand the evolution of $\mathbf{p}_{\mathsf{x}_n}$, we need to understand the structure of the n-fold matrix product $\mathbf{W}^n$, which is most easily revealed by representing $\mathbf{W}$ in terms of its *eigenvalues* and *eigenvectors*.

We illustrate the procedure in the case of our two-state example.

Example 4 (Two-State Markov Chain, Revisited Again)**.** In this example,

$$\mathbf{W} = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$

[5] A matrix whose columns and rows each sum to one is called *doubly stochastic*. It is a useful exercise to verify that transition matrices with this property correspond to Markov chains whose steady-state distributions are uniform.

As a reminder, we say that λ is an eigenvalue of the matrix $\mathbf{W}$, with corresponding eigenvector $\mathbf{u}$ if

$$\mathbf{u}\mathbf{W} = \lambda\mathbf{u} \tag{26}$$

Recall, too, that an eigenvector is never unique. For example, if $\mathbf{u}$ is an eigenvector associated with λ, so is $\mathbf{u}' = c\mathbf{u}$ for any c.

Since (26) can be equivalently written as

$$\mathbf{u}(\mathbf{W} - \lambda\mathbf{I}) = \mathbf{0}, \tag{27}$$

where

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

is the identity matrix, and where

$$\mathbf{0} = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$$

is the zero vector, we find the eigenvalues as the values for which

$$\det(\mathbf{W} - \lambda\mathbf{I}) = 0,$$

which is referred to as the *characteristic equation*, with $\det(\cdot)$ is the determinant of its argument.

In this case, the characteristic equation is

$$(1 - \alpha - \lambda)(1 - \beta - \lambda) - \alpha\beta = 0.$$

This equation is quadratic in λ, so it is satisfied by two values of λ. It is easy to verify that these values are

$$\lambda_1 = 1 \quad \text{and} \quad \lambda_2 = 1 - \alpha - \beta.$$

Moreover, substituting each of these eigenvalues into (27), and solving the resulting set of equations, we obtain suitable associated eigenvectors

$$\mathbf{u}_1 = \begin{bmatrix} \beta & \alpha \end{bmatrix} \quad \text{and} \quad \mathbf{u}_2 = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

As an aside, note that when normalized, $\mathbf{u}_1$, the eigenvector associated with the unity eigenvalue, is the vector representation $\mathbf{p}_x$ of the steady-state distribution, as we would expect.

Next note that we can stack the two equations

$$\mathbf{u}_1\mathbf{W} = \lambda_1\mathbf{u}_1 \quad \text{and} \quad \mathbf{u}_2\mathbf{W} = \lambda_2\mathbf{u}_2$$

together into the following form

$$\underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}}_{\mathbf{U}} \mathbf{W} = \begin{bmatrix} \lambda_1\mathbf{u}_1 \\ \lambda_1\mathbf{u}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}}_{\mathbf{U}}. \tag{28}$$

Hence, after left-multiplying both sides of (28) by $\mathbf{U}^{-1}$, we obtain

$$\mathbf{W} = \mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U}, \tag{29}$$

which is referred to as the *diagonalization* of $\mathbf{W}$

What makes the factorization (29) so useful is that we obtain n-fold products of $\mathbf{W}$ remarkably easily. In particular,

$$\begin{aligned}
\mathbf{W}^n &= (\mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U})^n & (30)\\
&= (\mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U})(\mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U})\cdots(\mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U}) & (31)\\
&= \mathbf{U}^{-1}\mathbf{\Lambda}^n\mathbf{U} & (32)\\
&= \frac{1}{\alpha+\beta}\begin{bmatrix}1 & -\alpha\\ 1 & \beta\end{bmatrix}\begin{bmatrix}1^n & 0\\ 0 & (1-\alpha-\beta)^n\end{bmatrix}\begin{bmatrix}\beta & \alpha\\ -1 & 1\end{bmatrix} & (33)\\
&= \frac{1^n}{\alpha+\beta}\begin{bmatrix}\beta & \alpha\\ \beta & \alpha\end{bmatrix} + \frac{(1-\alpha-\beta)^n}{\alpha+\beta}\begin{bmatrix}\alpha & -\alpha\\ -\beta & \beta\end{bmatrix}. & (34)
\end{aligned}$$

At this point, we note that the first term in (34) corresponds to the steady-state. In particular, for *any* initial state $\mathbf{p}_{\mathsf{x}_1}$, we have

$$\mathbf{p}_{\mathsf{x}_1}\frac{1^n}{\alpha+\beta}\begin{bmatrix}\beta & \alpha\\ \beta & \alpha\end{bmatrix} = \mathbf{p}_{\mathsf{x}}.$$

This is the term that corresponds to $\lambda_1 = 1$.

Moreover, the effect of the second term dies away with increasing n. Indeed,

$$\mathbf{p}_{\mathsf{x}_1}\frac{(1-\alpha-\beta)^n}{\alpha+\beta}\begin{bmatrix}\alpha & -\alpha\\ -\beta & \beta\end{bmatrix} = \frac{(1-\alpha-\beta)^n}{\alpha+\beta}\mathbf{p}_{\mathsf{x}_1}\begin{bmatrix}\alpha & -\alpha\\ -\beta & \beta\end{bmatrix}$$

is a vector whose magnitude vanishes with n since $|1-\alpha-\beta| < 1$. Thus, this is a transient contribution to the distribution p_{x_n}. This is the term that corresponds to $\lambda_2 = 1-\alpha-\beta$.

Hence, since $(1-\alpha-\beta)^n = e^{-n\ln(1-\alpha-\beta)^{-1}}$, we conclude that the distributions converge *exponentially* fast with n to the steady-state distribution. Moreover, the rate of exponential convergence is governed by the magnitude of the second eigenvalue.

The *mixing time* for a Markov chain is the time duration until the distribution is *close* to the steady-state one, by some useful measure. Hence the mixing time is determined by this second eigenvalue. In particular, the closer α and β are to 1/2, the faster the mixing, and when they are both close to 0 or 1, the mixing is very slow.

The behavior in this example is representative. Specifically, for an arbitrary Markov chain of m states, there are m eigenvalues, some possibly repeated. The largest eigenvalue is always one, and the associated eigenvector, after normalization, gives our steady-state distribution. If, e.g., $\mathbf{W}$ has strictly positive entries, the unity eigenvalue is not repeated, so steady-state distribution is unique. The eigenvalue with the second largest magnitude is the one that dominates the convergence to the steady-state, and thus controls the mixing time.

In practice, for even moderately sized state alphabets, determining this second largest eigenvalue is often difficult, and one often has to resort to bounds. Even then, developing useful bounds requires considerable creativity.

A.4 Reversible Markov Chains

A *reversible* Markov chain has the property that when operating in its steady-state, the chain look the same (probabilistically) whether time runs forwards or backwards, i.e., for all n,

$$p_{\mathsf{x}_1,\mathsf{x}_2,\ldots,\mathsf{x}_n}(x_1, x_2, \ldots, x_n) = p_{\mathsf{x}_1,\mathsf{x}_2,\ldots,\mathsf{x}_n}(x_n, x_{n-1}, \ldots, x_1).$$

A Markov chain reversible if and only if its the steady-state distributions satisfy not only the global balance equations (20), but the following *detailed balance* equations:

$$p_{\mathsf{x}}(a')\, W(a|a') = p_{\mathsf{x}}(a)\, W(a'|a). \tag{35}$$

By summing both sides of (35) over a', we immediately obtain the global balance equations, so indeed detailed balance implies global balance. This additional structure is often useful.