Contents

1 Markov Chains

The study of Markov chains is a classical subject with many applications such as Markov Chain Monte Carlo techniques for integrating multivariate probability distributions over complex volumes. An important recent application is in defining the pagerank of pages on the World Wide Web by their stationary probabilities.

A Markov chain has a finite set of *states*. For each pair x and y of states, there is a probability p_{xy} of going from state x to state y where for each x, $\sum_y p_{xy} = 1$. A random walk in the Markov chain consists of a sequence of states starting at some state x_0. In state x, the next state y is selected randomly with probability p_{xy}. The starting probability distribution puts a mass of one on the start state x_0 and zero on every other state. More generally, one could start with any probability distribution $\mathbf{p}$, where $\mathbf{p}$ is a row vector with non-negative components summing to one, with p_i being the probability of starting in state i. The probability of being at state j at time $t+1$ is the sum over each state i of being at i at time t and taking the transition from i to j. Let $\mathbf{p^{(t)}}$ be a row vector with a component for each state specifying the probability mass of the state at time t and let $\mathbf{p^{(t+1)}}$ be the row vector of probabilities at time $t+1$. In matrix notation

$$\mathbf{p^{(t)}}P = \mathbf{p^{(t+1)}}.$$

Many real-world situations can be modeled as Markov chains. At any time, the only information about the chain is the current state, not how the chain got there. At the next unit of time the state is a random variable whose distribution depends only on the current state. A gambler's assets can be modeled as a Markov chain where the current state is the amount of money the gambler has on hand. The model would only be valid if the next state does not depend on past states, only on the current one. Human speech has been modeled as a Markov chain, where the state represents either the last syllable (or the last several syllables) uttered. The reader may consult sources on Markov chains for other examples; our discussion here focuses on the theory behind them.

A Markov chain can be represented by a directed graph with a vertex representing each state and an edge labeled p_{ij} from vertex i to vertex j if $p_{ij} > 0$. We say that the Markov chain is *strongly connected* if there is a directed path from each vertex to every other vertex. The matrix P of the p_{ij} is called the *transition probability matrix* of the chain.

A fundamental property of a Markov chain is that in the limit the long-term average probability of being in a particular state is independent of the start state or an initial probability distribution over states provided that the directed graph is strongly connected. This is the Fundamental Theorem of Markov chains which we now prove.

1.1 Stationary Distribution

Suppose after t steps of the random walk, the probability distribution is $\mathbf{p^{(t)}}$. Define the *long-term probability distribution* $\mathbf{a^{(t)}}$ by

$$\mathbf{a^{(t)}} = \frac{1}{t}\left(\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right).$$

The next theorem proves that the long-term probability distribution of a strongly connected Markov chain converges to a unique probability vector. This does not mean that the probability distribution of the random walk converges. This would require an additional condition called aperiodic.

Theorem 1.1 (Fundamental Theorem of Markov chains) *If the Markov chain is strongly connected, there is a unique probability vector* $\boldsymbol{\pi}$ *satisfying* $\boldsymbol{\pi}P = \boldsymbol{\pi}$*. Moreover, for any starting distribution,* $\lim_{t\to\infty} \mathbf{a^{(t)}}$ *exists and equals* $\boldsymbol{\pi}$*.*

Proof:

$$\begin{aligned}
\mathbf{a^{(t)}}P - \mathbf{a^{(t)}} &= \frac{1}{t}\left[\mathbf{p^{(0)}}P + \mathbf{p^{(1)}}P + \cdots + \mathbf{p^{(t-1)}}P\right] - \frac{1}{t}\left[\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right] \\
&= \frac{1}{t}\left[\mathbf{p^{(1)}} + \mathbf{p^{(2)}} + \cdots + \mathbf{p^{(t)}}\right] - \frac{1}{t}\left[\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right] \\
&= \frac{1}{t}\left(\mathbf{p^{(t)}} - \mathbf{p^{(0)}}\right).
\end{aligned}$$

Thus, $\mathbf{b^{(t)}} = \mathbf{a^{(t)}}P - \mathbf{a^{(t)}}$ satisfies $|\mathbf{b^{(t)}}| \leq \frac{2}{t} \to 0$, as $t \to \infty$. Letting A be the $n \times (n+1)$ matrix $[P - I\ ,\ \mathbf{1}]$ obtained by augmenting the matrix $P - I$ with an additional column of ones. Then $\mathbf{a^{(t)}}A = [\mathbf{b^{(t)}}\ ,\ 1]$. The matrix A has rank n since each row sum in P is 1 and hence row sums in $P - I$ are all 0. Thus $A\begin{pmatrix}\mathbf{1}\\0\end{pmatrix} = 0$. If the rank of A is less than n, there is a vector $\mathbf{w}$ perpendicular to $\mathbf{1}$ and scalar α so that $(P-I)\mathbf{w} = \alpha\mathbf{1}$ or $P\mathbf{w} - \alpha\mathbf{1} = \mathbf{w}$. If $\alpha > 0$, then for the i with maximum value of w_i, $\mathbf{w}w_i$ is a convex combination of some w_j, all at most w_i minus α, a contradiction. Similarly for $\alpha < 0$. So assume $\alpha = 0$. For the i with maximum w_i, if for some j, $p_{ij} > 0$, then $w_j = w_i$. Otherwise, $(P\mathbf{w})_i$ would be less than w_i. Now suppose S is the set of k with w_k equal to the maximum value. $\bar{S}$ is not empty since $\sum_k w_k = 0$. Connectedness implies that there exist $k \in S, l \in \bar{S}$ with $p_{kl} > 0$, which is a contradiction. So A has rank n and the $n \times n$ submatrix B of A consisting of all its columns except the first is invertible. Let $\mathbf{c^{(t)}}$ be obtained from $\mathbf{b^{(t)}}$ by removing the first entry. Then, $\mathbf{a^{(t)}} = [\mathbf{c^{(t)}}\ ,\ 1]B^{-1} \to [\mathbf{0}\ ,\ 1]B^{-1}$. We have the theorem with $\boldsymbol{\pi} = [\mathbf{0}\ ,\ 1]B^{-1}$. ■

The vector $\boldsymbol{\pi}$ is called the *stationary probability distribution* of the Markov chain. The equations $\boldsymbol{\pi}P = \boldsymbol{\pi}$ expanded say that for every i,

$$\sum_j \pi_j p_{ji} = \pi_i.$$

Thus, executing one step of the Markov Chain starting with the distribution $\boldsymbol{\pi}$ results in the same distribution. Of course the same conclusion holds for any number of steps. Hence the name stationary distribution, sometimes called the steady state distribution.

1.2 Electrical Networks and Random Walks

In the next few sections, we study a special class of Markov chains derived from electrical networks. These include Markov chains on undirected graphs where one of the edges incident to the current vertex is chosen uniformly at random and the walk proceeds to the vertex at the other end of the edge. There are nice analogies between such Markov chains and certain electrical quantities.

An electrical network is a connected, undirected graph in which each edge xy has a resistance $r_{xy} > 0$. In what follows, it is easier to deal with conductance defined as the reciprocal of resistance, $c_{xy} = \frac{1}{r_{xy}}$, rather then resistance. Associated with an electrical network is a Markov chain on the underlying graph defined by assigning a probability $p_{xy} = \frac{c_{xy}}{c_y}$ to the edge (x, y) incident to a vertex, where the normalizing constant c_x equals $\sum\limits_{y} c_{xy}$. Note that although c_{xy} equals c_{yx}, the probabilities p_{xy} and p_{yx} may not be equal due to the required normalization so that the probabilities at each vertex sum to one. Thus, the matrix P may not be symmetric. We shall soon see that there is a relationship between current flowing in an electrical network and a random walk on the underlying graph.

Denote by P the matrix whose xy^{th} entry p_{xy} is the probability of a transition from x to y. The matrix P is called the *transition probability matrix*. Suppose a random walk starts at a vertex x_0. At the start, the probability mass is one at x_0 and zero at all other vertices. At time one, for each vertex y, the probability of being at y is the probability, $p_{x_0 y}$, of going from x_0 to y.

If the underlying electrical network is connected, then the Markov chain is strongly connected and has a stationary probability distribution. We claim that the stationary probability is given by $f_x = \frac{c_x}{c}$ where $c = \sum\limits_{x} c_x$. By Theorem 1.1, it suffices to check that $\mathbf{f}P = \mathbf{f}$:

$$(\mathbf{f}P)_x = \sum_y \frac{c_y}{c} \frac{c_{yx}}{c_y} = \sum_y \frac{c_{xy}}{c} = \frac{c_x}{c} = f_x.$$

Note that if each edge has resistance one, then the value of $c_x = \sum\limits_{y} c_{xy}$ is d_x where d_x is the degree of x. In this case, $c = \sum\limits_{x} c_x$ equals $2m$ where m is the total number of edges and the stationary probability is $\frac{1}{2m}\left(d_1, d_2, \ldots, d_n\right)$. This means that for undirected graphs, the stationary probability of each vertex is proportional to its degree and if the walk starts with the stationary distribution, every edge is traversed in each direction with the same probability of $\frac{1}{2m}$.

A random walk associated with an electrical network has the important property that given the stationary probability $\mathbf{f}$, the probability $f_x p_{xy}$ of traversing the edge xy from vertex x to vertex y is the same as the probability $f_y p_{yx}$ of traversing the edge in the reverse direction from vertex y to vertex x. This follows from the manner in which probabilities were assigned and the fact that the conductance c_{xy} equals c_{yx}.

$$f_x p_{xy} = \frac{c_x}{c}\frac{c_{xy}}{c_x} = \frac{c_{xy}}{c} = \frac{c_{yx}}{c} = \frac{c_y}{c}\frac{c_{yx}}{c_y} = f_y p_{yx}.$$

Harmonic functions

Harmonic functions are useful in developing the relationship between electrical networks and random walks on undirected graphs. Given an undirected graph, designate a nonempty set of vertices as boundary vertices and the remaining vertices as interior vertices. A harmonic function g on the vertices is one in which the value of the function at the boundary vertices is fixed to some boundary condition and the value of g at any interior vertex x is a weighted average of the values at all the adjacent vertices y, where the weights p_{xy} sum to one over all y. Thus, if $g_x = \sum\limits_y g_y p_{xy}$ at every interior vertex x, then g is harmonic. From the fact that $\mathbf{f}P = \mathbf{f}$, it follows that the function $g_x = \frac{f_x}{c_x}$ is harmonic:

$$\begin{aligned} g_x = \tfrac{f_x}{c_x} &= \tfrac{1}{c_x}\sum_y f_y p_{yx} = \tfrac{1}{c_x}\sum_y f_y \tfrac{c_{yx}}{c_y} \\ &= \tfrac{1}{c_x}\sum_y f_y \tfrac{c_{xy}}{c_y} = \sum_y \tfrac{f_y}{c_y}\tfrac{c_{xy}}{c_x} = \sum_y g_y p_{xy}. \end{aligned}$$

A harmonic function on a connected graph takes on its maximum and minimum on the boundary. Suppose not. Let S be the set of interior vertices at which the maximum value is attained. Since S contains no boundary vertices, $\bar{S}$ is nonempty. Connectedness implies that there is at least one edge (x, y) with $x \in S$ and $y \in \bar{S}$. But then the value of the function at x is the average of the value at its neighbors, all of which are less than or equal to the value at x and the value at y is strictly less, a contradiction. The proof for the minimum value is identical.

There is at most one harmonic function satisfying a given set of equations and boundary conditions. For suppose there were two solutions $f(x)$ and $g(x)$. The difference of two solutions is itself harmonic. Since $h(x) = f(x) - g(x)$ is harmonic and has value zero on the boundary, by the maximum principle it has value zero everywhere. Thus $f(x) = g(x)$.

The analogy between electrical networks and random walks

There are important connections between random walks on undirected graphs and electrical networks. Choose two vertices a and b. For reference purposes let the voltage

v_b equal zero. Attach a current source between a and b so that the voltage v_a equals one. Fixing the voltages at v_a and v_b induces voltages at all other vertices along with a current flow through the edges of the network. The analogy between electrical networks and random walks is the following. Having fixed the voltages at the vertices a and b, the voltage at an arbitrary vertex x equals the probability of a random walk starting at x reaching a before reaching b. If the voltage v_a is adjusted so that the current flowing into vertex a is one, then the current flowing through an edge is the net frequency in which a random walk from a to b traverses the edge.

Probabilistic interpretation of voltages

Before showing that the voltage at an arbitrary vertex x equals the probability of a random walk from x reaching a before reaching b, we first show that the voltages form a harmonic function. Let x and y be adjacent vertices and let i_{xy} be the current flowing through the edge from x to y. By Ohm's law,

$$i_{xy} = \frac{v_x - v_y}{r_{xy}} = (v_x - v_y)c_{xy}.$$

By Kirchoff's Law the currents flowing out of each vertex sum to zero.

$$\sum_y i_{xy} = 0$$

Replacing currents in the above sum by the voltage difference times the conductance yields

$$\sum_y (v_x - v_y)c_{xy} = 0$$

or

$$v_x \sum_y c_{xy} = \sum_y v_y c_{xy}.$$

Observing that $\sum\limits_y c_{xy} = c_x$ and that $p_{xy} = \frac{c_{xy}}{c_x}$, yields $v_x c_x = \sum\limits_y v_y p_{xy} c_x$. Hence, $v_x = \sum\limits_y v_y p_{xy}$. Thus, the voltage at each vertex x is a weighted average of the voltages at the adjacent vertices. Hence the voltages are harmonic.

Now let p_x be the probability that a random walk starting at vertex x reaches a before b. Clearly $p_a = 1$ and $p_b = 0$. Since $v_a = 1$ and $v_b = 0$, it follows that $p_a = v_a$ and $p_b = v_b$. Furthermore, the probability of the walk reaching a from x before reaching b is the sum over all y adjacent to x of the probability of the walk going from x to y and then reaching a from y before reaching b. That is

$$p_x = \sum_y p_{xy} p_y.$$

Hence, p_x is the same harmonic function as the voltage function v_x and $\mathbf{v}$ and $\mathbf{p}$ satisfy the same boundary conditions (a and b form the boundary). Thus, they are identical functions. The probability of a walk starting at x reaching a before reaching b is the voltage v_x.

Probabilistic interpretation of current

In a moment we will set the current into the network at a to have some value which we will equate with one random walk. We will then show that the current i_{xy} is the net frequency with which a random walk from a to b goes through the edge xy before reaching b. Let u_x be the expected number of visits to vertex x on a walk from a to b before reaching b. Clearly $u_b = 0$. Since every time the walk visits x, x not equal to a, it must come to x from some vertex y, the number of visits to x before reaching b is the sum over all y of the number of visits u_y to y before reaching b times the probability p_{yx} of going from y to x. Thus

$$u_x = \sum_y u_y p_{yx} = \sum_y u_y \frac{c_x p_{xy}}{c_y}$$

and hence $\frac{u_x}{c_x} = \sum\limits_y \frac{u_y}{c_y} p_{xy}$. It follows that $\frac{u_x}{c_x}$ is harmonic (with a and b as the boundary). Now, $\frac{u_b}{c_b} = 0$. Setting the current into a to one, fixed the value of v_a. Adjust the current into a so that v_a equals $\frac{u_a}{c_a}$. Since $\frac{u_x}{c_x}$ and v_x satisfy the same harmonic conditions, they are the same harmonic function. Let the current into a correspond to one walk. Note that if our walk starts at a and ends at b, the expected value of the difference between the number of times the walk leaves a and enters a must be one and thus the amount of current into a corresponds to one walk.

Next we need to show that the current i_{xy} is the net frequency with which a random walk traverses edge xy.

$$i_{xy} = (v_x - v_y)c_{xy} = \left(\frac{u_x}{c_x} - \frac{u_y}{c_y}\right) c_{xy} = u_x \frac{c_{xy}}{c_x} - u_y \frac{c_{xy}}{c_y} = u_x p_{xy} - u_y p_{yx}$$

The quantity $u_x p_{xy}$ is the expected number of times the edge xy is traversed from x to y and the quantity $u_y p_{yx}$ is the expected number of times the edge xy is traversed from y to x. Thus, the current i_{xy} is the expected net number of traversals of the edge xy from x to y.

Effective Resistance and Escape Probability

Set $v_a = 1$ and $v_b = 0$. Let i_a be the current flowing into the network at vertex a and out at vertex b. Define the *effective resistance* r_{eff} between a and b to be $r_{eff} = \frac{v_a}{i_a}$ and the *effective conductance* c_{eff} to be $c_{eff} = \frac{1}{r_{eff}}$. Define the *escape probability*, p_{escape}, to be the probability that a random walk starting at a reaches b before returning to a. We now show that the escape probability is $\frac{c_{eff}}{c_a}$.

$$i_a = \sum_y (v_a - v_y)c_{ay}$$

Since $v_a = 1$,

$$\begin{aligned} i_a &= \sum_y (1 - v_y) \frac{c_{ay}}{c_a} c_a \\ &= c_a \left[\sum_y \frac{c_{ay}}{c_a} - \sum_y v_y \frac{c_{ay}}{c_a} \right] \\ &= c_a \left[1 - \sum_y p_{ay} v_y \right]. \end{aligned}$$

For each y adjacent to the vertex a, p_{ay} is the probability of the walk going from vertex a to vertex y. v_y is the probability of a walk starting at y going to a before reaching b, as was just argued. Thus, $\sum\limits_y p_{ay} v_y$ is the probability of a walk starting at a returning to a before reaching b and $1 - \sum\limits_y p_{ay} v_y$ is the probability of a walk starting at a reaching b before returning to a. Thus $i_a = c_a p_{escape}$. Since $v_a = 1$ and $c_{eff} = \frac{i_a}{v_a}$, it follows that $c_{eff} = i_a$. Thus $c_{eff} = c_a p_{escape}$ and hence $p_{escape} = \frac{c_{eff}}{c_a}$.

For a finite graph the escape probability will always be nonzero. Now consider an infinite graph such as a lattice and a random walk starting at some vertex a. Form a series of finite graphs by merging all vertices at distance d or greater from a into a single vertex b for larger and larger values of d. The limit of p_{escape} as d goes to infinity is the probability that the random walk will never return to a. If $p_{escape} \to 0$, then eventually any random walk will return to a. If $p_{escape} \to q$ where $q > 0$, then a fraction of the walks never return. Thus, the escape probability terminology.

1.3 Random Walks on Undirected Graphs

We now focus our discussion on random walks on undirected graphs with uniform edge weights. At each vertex, the random walk is equally likely to take any edge. This corresponds to an electrical network in which all edge resistances are one. Assume the graph is connected. If it is not, the analysis below can be applied to each connected component separately. We consider questions such as what is the expected time for a random walk starting at a vertex x to reach a target vertex y, what is the expected time until the random walk returns to the vertex it started at, and what is the expected time to reach every vertex?

Hitting time

The *hitting time* h_{xy}, sometimes called *discovery time*, is the expected time of a random walk starting at vertex x to reach vertex y. Sometimes a more general definition is given where the hitting time is the expected time to reach a vertex y from a start vertex selected at random from some given probability distribution.

One interesting fact is that adding edges to a graph may either increase or decrease h_{xy} depending on the particular situation. An edge can shorten the distance from x to y thereby decreasing h_{xy} or the edge could increase the probability of a random walk going to some far off portion of the graph thereby increasing h_{xy}. Another interesting fact is that hitting time is not symmetric. The expected time to reach a vertex y from a vertex x in an undirected graph may be radically different from the time to reach x from y.

We start with two technical lemmas. The first lemma states that the expected time to traverse a chain of n vertices is $\Theta\left(n^2\right)$.

Lemma 1.2 *The expected time for a random walk starting at one end of a chain of n vertices to reach the other end is $\Theta\left(n^2\right)$.*

Proof: Consider walking from vertex 1 to vertex n in a graph consisting of a single path of n vertices. Let h_{ij}, $i < j$, be the hitting time of reaching j starting from i. Now $h_{12} = 1$ and

$$h_{i,i+1} = \tfrac{1}{2} \times 1 + \tfrac{1}{2}\left(1 + h_{i-1,i} + h_{i,i+1}\right) \quad 2 \le i \le n-1.$$

Solving for $h_{i,i+1}$ yields the recurrence

$$h_{i,i+1} = 2 + h_{i-1,i}.$$

Solving the recurrence yields

$$h_{i,i+1} = 2i - 1.$$

To get from 1 to n, go from 1 to 2, 2 to 3, etc. Thus

$$\begin{aligned} h_{1,n} &= \sum_{i=1}^{n-1} h_{i,i+1} = \sum_{i=1}^{n-1} (2i-1) \\ &= 2\sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} 1 \\ &= 2\frac{n(n-1)}{2} - (n-1) \\ &= (n-1)^2 . \end{aligned}$$

■

The next lemma shows that the expected time spent at vertex i by a random walk from vertex 1 to vertex n in a chain of n vertices is $2(i-1)$ for $2 \le i \le n-1$.

Lemma 1.3 *Consider a random walk from vertex 1 to vertex n in a chain of n vertices. Let t(i) be the expected time spent at vertex i. Then*

$$t(i) = \begin{cases} n-1 & i = 1 \\ 2(n-i) & 2 \le i \le n-1 \\ 1 & i = n. \end{cases}$$

Proof: Now $t\left(n\right)=1$ since the walk stops when it reaches vertex n. Half of the time when the walk is at vertex $n-1$ it goes to vertex n. Thus $t\left(n-1\right)=2$. For $3\leq i\leq n-1$, $t\left(i\right)=\frac{1}{2}\left[t\left(i-1\right)+t\left(i+1\right)\right]$ and $t\left(1\right)$ and $t\left(2\right)$ satisfy $t\left(1\right)=\frac{1}{2}t\left(2\right)+1$ and $t\left(2\right)=t\left(1\right)+\frac{1}{2}t\left(3\right)$. Solving for $t(i+1)$ for $3\leq i\leq n-1$ yields

$$t(i+1)=2t(i)-t(i-1)$$

which has solution $t(i)=2(n-i)$ for $3\leq i\leq n-1$. Then solving for $t(2)$ and $t(1)$ yields $t\left(2\right)=2\left(n-2\right)$ and $t\left(1\right)=n-1$. Thus, the total time spent at vertices is

$$n-1+2\left(1+2+\cdots+n-2\right)+1=n-1+(n-1)(n-2)+1=\left(n-1\right)^{2}+1$$

which is one more than h_{1n} and thus is correct. ■

Next we show that adding edges to a graph might either increase or decrease the hitting time h_{xy}. Consider the graph consisting of a single path of n vertices. Add edges to this graph to get the graph in Figure 1.1 consisting of a clique of size $n/2$ connected to a path of $n/2$ vertices. Then add still more edges to get a clique of size n. Let x be the vertex at the midpoint of the original path and let y be the other endpoint of the path consisting of $n/2$ vertices as shown in Figure 1.1. In the first graph consisting of a single path of length n, $h_{xy}=\Theta\left(n^{2}\right)$. In the second graph consisting of a clique of size $n/2$ along with a path of length $n/2$, $h_{xy}=\Theta\left(n^{3}\right)$. To see this latter statement, note that starting at x, the walk will go down the chain towards y and return to x n times on average before reaching y for the first time. Each time the walk in the chain returns to x, with probability $(n-1)/n$ it enters the clique and thus on average enters the clique $\Theta(n)$ times before starting down the chain again. Each time it enters the clique, it spends $\Theta(n)$ time in the clique before returning to x. Thus, each time the path returns to x from the chain it spends $\Theta(n^{2})$ time in the clique before starting down the chain towards y for a total expected time that is $\Theta(n^{3})$ before reaching y. In the third graph, which is the clique of size n, $h_{xy}=\Theta\left(n\right)$. Thus, adding edges first increased h_{xy} from n^{2} to n^{3} and then decreased it to n.

Hitting time is not symmetric even in the case of undirected graphs. In the graph of Figure 1.1, the expected time, h_{xy}, of a random walk from x to y, where x is the vertex of attachment and y is the other end vertex of the chain, is $\Theta(n^{3})$. However, h_{yx} is $\Theta(n^{2})$.

Next we ask what is the maximum that the hitting time could be. We first show that if vertices x and y are connected by an edge, then the expected time, h_{xy}, of a random walk from x to y plus the expected time, h_{yx}, from y to x is at most twice the number of edges.

Lemma 1.4 *If vertices x and y are connected by an edge, then $h_{xy}+h_{yx}\leq 2m$ where m is the number of edges in the graph.*

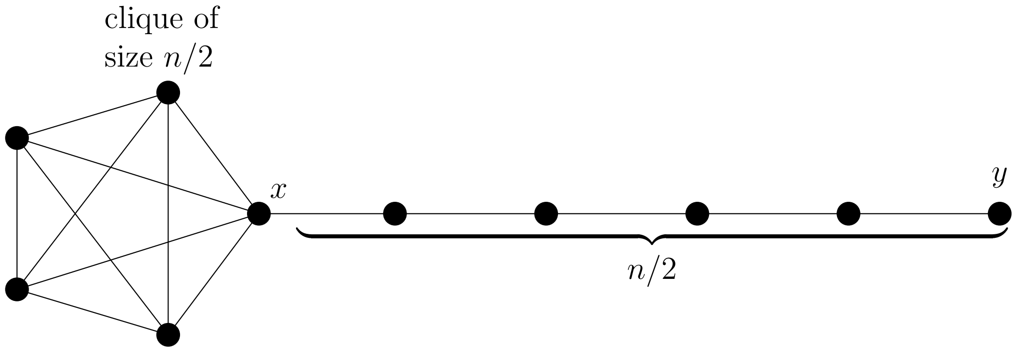


Figure 1.1: Illustration that adding edges to a graph can either increase or decrease hitting time.

Proof: In a random walk on an undirected graph starting in the steady state, the probability of traversing any edge in either direction is $1/(2m)$. This is because for any edge (u, v), the probability of being at u (in the steady state) is $d_u/(2m)$ and the probability of selecting the edge (u, v) is $1/d_u$. Hence, the probability of traversing the edge (u, v) is $1/(2m)$ implying that the expected time between traversals of the edge (x, y) from x to y is $2m$. Thus, if we traverse edge (x, y), the expected time to traverse a path from y back to x and then traverse the edge (x, y) again is $2m$. But since a random walk is a memory less process, we can drop the condition that we started by traversing the edge (x, y). Hence the expected time from y to x and back to y is at most $2m$. Note that the path went from y to x and then may have returned to x several times before going through the edge (x, y). Thus, the less than or equal sign in the statement of the lemma since the path have gone from y to x to y without going through the edge (x, y). ∎

Notice that the proof relied on the fact that there was an edge from x to y and thus the theorem is not necessarily true for arbitrary x and y. When x and y are not connected by an edge consider a path from x to y. The path is of length at most n. Consider the time it takes to reach each vertex on the path in the order they appear. Since the vertices on the path are connected by edges, the expected time to reach the next vertex on the path is at most twice the number of edges in the graph by the above theorem. Thus, the total expected time is $\Theta\left(n^3\right)$. This result is asymptotically tight since the bound is met by the graph of Figure 1.1 consisting of a clique of size $n/2$ and a path of length $n/2$.

Commute time

The *commute time*, commute(x, y), is the expected time of a random walk starting at x reaching y and then returning to x. Think of going from home to office and returning home.

Theorem 1.5 *Given an undirected graph, consider the electrical network where each edge of the graph is replaced by a one ohm resistor. Given vertices x and y, the commute time,* *commute*(x, y), *equals* $2mr_{xy}$ *where* r_{xy} *is the effective resistance from* x *to* y *and* m *is the number of edges in the graph.*

Proof: Insert at each vertex i a current equal to the degree d_i of vertex i. The total current inserted is $2m$ where m is the number of edges. Extract from a specific vertex j all of this $2m$ current. Let v_{ij} be the voltage difference from i to j. The current into i divides into the d_i resistors at node i. The current in each resistor is proportional to the voltage across it. Let k be a vertex adjacent to i. Then the current through the resistor between i and k is $v_{ij} - v_{kj}$, the voltage drop across the resister. The sum of the currents out of i through the resisters must equal d_i, the current injected into i.

$$d_i = \sum_{\substack{k \text{ adj} \\ \text{to } i}} (v_{ij} - v_{kj})$$

Noting that v_{ij} does not depend on k, write

$$d_i = d_i v_{ij} - \sum_{\substack{k \text{ adj} \\ \text{to } i}} v_{kj}.$$

Solving for v_{ij}

$$v_{ij} = 1 + \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i} v_{kj} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i}(1 + v_{kj}). \tag{1.1}$$

Now the expected time from i to j is the average time over all paths from i to k adjacent to i and then on from k to j. This is given by

$$h_{ij} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i}(1 + h_{kj}). \tag{1.2}$$

Subtracting (1.2) from (1.1), gives $v_{ij} - h_{ij} = \sum\limits_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i}(v_{kj} - h_{kj})$. Thus, the function $v_{ij} - h_{ij}$ is harmonic. Designate vertex j as the only exterior vertex. The value of $v_{ij} - h_{ij}$ at j, namely $v_{jj} - h_{jj}$, is zero, since both v_{jj} and h_{jj} are zero. So the function $v_{ij} - h_{ij}$ must be zero everywhere. Thus, the voltage v_{ij} equals the expected time h_{ij} from i to j.

To complete the proof, note that $h_{ij} = v_{ij}$ is the voltage from i to j when currents are inserted at all nodes in the graph and extracted at node j. If the current is extracted from i instead of j, then the voltages change and $v_{ji} = h_{ji}$ in the new setup. Finally, reverse all currents in this latter step. The voltages change again and for the new voltages $-v_{ji} = h_{ji}$. Since $-v_{ji} = v_{ij}$, we get $h_{ji} = v_{ij}$.

Thus, when a current is inserted at each node equal to the degree of the node and the current is extracted from j, the voltage v_{ij} in this set up equals h_{ij}. When we extract the current from i instead of j and then reverse all currents, the voltage v_{ij} in this new set up equals h_{ji}. Now, superpose both situations (i.e., add all the currents and voltages). By linearity, for the resulting v_{ij}, $v_{ij} = h_{ij} + h_{ji}$. All currents cancel except the $2m$ amps injected at i and withdrawn at j. Thus, $2mr_{ij} = v_{ij} = h_{ij} + h_{ji} = \text{commute}(i, j)$. Thus, $\text{commute}(i, j) = 2mr_{ij}$. ■

Note that Lemma 1.4 also follows from Theorem 1.5 since the effective resistance r_{uv} is less than or equal to 1 when u and v are connected by an edge.

Corollary 1.6 *For any n vertex graph and for any vertices x and y, the commute time, commute(x, y), is less than or equal to n^3.*

Proof: By Theorem 1.5 the commute time is given by the formula $\text{commute}(x, y) = 2mr_{xy}$ where m is the number of edges. In an n vertex graph there exists a path from x to y of length at most n. This implies $r_{xy} \leq n$ since the resistance can not be greater than that of any path from x to y. Since the number of edges is at most $\binom{n}{2}$

$$\text{commute}(x, y) = 2mr_{xy} \leq 2\binom{n}{2} n \cong n^3.$$

■

Again adding edges to a graph may increase or decrease the commute time. To see this, consider the graph consisting of a chain of n vertices, the graph of Figure 1.1, and the clique on n vertices.

Cover times

The *cover time* $\text{cover}(x, G)$ is the expected time of a random walk starting at vertex x in the graph G to reach each vertex at least once. We write $\text{cover}(x)$ when G is understood. The cover time of an undirected graph G, denoted $\text{cover}(G)$, is

$$\text{cover}(G) = \max_x \text{cover}(x, G).$$

For cover time of an undirected graph, increasing the number of edges in the graph may increase or decrease the cover time depending on the situation. Again consider three graphs, a chain of length n which has cover time $\Theta(n^2)$, the graph in Figure 1.1 which has cover time $\Theta(n^3)$, and the complete graph on n vertices which has cover time $\Theta(n \log n)$. Adding edges to the chain of length n to create the graph in Figure 1.1 increases the cover time from n^2 to n^3 and then adding even more edges to obtain the complete graph reduces the cover time to $n \log n$.

Note: The cover time of a clique is $n \log n$ since that is the time to select every integer out of n integers with high probability, drawing integers at random. This is called the coupon collector problem. The cover time for a straight line is $\Theta(n^2)$ since it is the same as the hitting time. For the graph in Figure 1.1, the cover time is $\Theta(n^3)$ since one takes the maximum over all start states and $\text{cover}(x, G) = \Theta\left(n^3\right)$.

Theorem 1.7 *Let G be a connected graph with n vertices and m edges. The time for a random walk to cover all vertices of the graph G is bounded above by $2m(n-1)$.*

Proof: Consider a depth first search (dfs) of the graph G starting from vertex z and let T be the resulting dfs spanning tree of G. The dfs covers every vertex. Consider the expected time to cover every vertex in the order visited by the depth first search. Clearly this bounds the cover time of G starting from vertex z.

$$\text{cover}\,(z, G) \leq \sum_{(x,y)\in T} h_{xy}.$$

If (x, y) is an edge in T, then x and y are adjacent and thus Lemma 1.4 implies $h_{xy} \leq 2m$. Since there are $n-1$ edges in the dfs tree and each edge is traversed twice, once in each direction, $\text{cover}(z) \leq 2m(n-1)$. Since this holds for all starting vertices z, $\text{cover}(G) \leq 2m(n-1)$ ■

The theorem gives the correct answer of n^3 for the $n/2$ clique with the $n/2$ tail. It gives an upper bound of n^3 for the n-clique where the actual cover time is $n \log n$.

Let r_{xy} be the effective resistance from x to y. Define the resistance $r(G)$ of a graph G by $r(G) = \max\limits_{x,y}(r_{xy})$.

Theorem 1.8 *Let G be an undirected graph with m edges. Then the cover time for G is bounded by the following inequality*

$$mr(G) \leq \textit{cover}(G) \leq 2e^3 mr(G) \ln n + n$$

where e=2.71 is Euler's constant and $r(G)$ is the resistance of G.

Proof: By definition $r(G) = \max\limits_{x,y}(r_{xy})$. Let u and v be the vertices of G for which r_{xy} is maximum. Then $r(G) = r_{uv}$. By Theorem 1.5, $\text{commute}(u, v) = 2mr_{uv}$. Hence $mr_{uv} = \frac{1}{2}\text{commute}(u, v)$. Clearly the commute time from u to v and back to u is less than twice the $\max(h_{uv}, h_{vu})$ and $\max(h_{uv}, h_{vu})$ is clearly less than the cover time of G. Putting these facts together

$$mr(G) = mr_{uv} = \tfrac{1}{2}\text{commute}(u, v) \leq \max(h_{uv}, h_{vu}) \leq \text{cover}(G).$$

For the second inequality in the theorem, by Theorem 1.5, for any x and y $\text{commute}(x, y)$ equals $2mr_{xy}$ implying $h_{xy} \leq 2mr(G)$. By the Markov inequality, since the expected value of h_{xy} is less than $2mr(G)$, the probability that y is not reached from x in $2mr(G)e^3$ steps is at most $\frac{1}{e^3}$. Thus, the probability that a vertex y has not been reached in $2e^3mr(G)\log n$ steps is at most $\frac{1}{e^3}^{\ln n} = \frac{1}{n^3}$ because a random walk of length $2e^3mr(G)\log n$ is a sequence of $\log n$ independent random walks, each of length $2e^3mr(G)$. Suppose after a walk of $2e^3mr(G)\log n$ steps, vertices $v_1, v_2, \ldots, v_l$ where not reached. Walk until v_1 is reached, then v_2, etc. By Corollary 1.6 the expected time for each of these is n^3, but since each happens only with probability $1/n^3$, we effectively take $O(1)$ time per v_i, for a total time of at most n. ■

Return time

The *return time* is the expected time of a walk starting at x returning to x. We explore this quantity later.

1.4 Random Walks in Euclidean Space

Many physical processes such as Brownian motion are modeled by random walks. Random walks in Euclidean d-space consisting of fixed length steps parallel to the coordinate axes are really random walks on a d-dimensional lattice and are a special case of random walks on graphs. In a random walk on a graph, at each time unit an edge from the current vertex is selected at random and the walk proceeds to the adjacent vertex. We begin by studying random walks on lattices.

Random walks on lattices

We now apply the analogy between random walks and current to lattices. Consider a random walk on a finite segment $-n, \ldots, -1, 0, 1, 2, \ldots, n$ of a one dimensional lattice starting from the origin. Is the walk certain to return to the origin or is there some probability that it will escape, i.e., reach the boundary before returning? The probability of reaching the boundary before returning to the origin is called the escape probability. We shall be interested in this quantity as n goes to infinity.

Convert the lattice to an electrical network by replacing each edge with a one ohm resister. Then the probability of a walk starting at the origin reaching n or $-n$ before returning to the origin is the escape probability given by

$$p_{escape} = \frac{c_{eff}}{c_a}$$

where c_{eff} is the effective conductance between the origin and the boundary points and c_a is the sum of the conductance's at the origin. In a d-dimensional lattice, $c_a = 2d$ assuming that the resistors have value one. For the d-dimensional lattice

$$p_{escape} = \frac{1}{2d\, r_{eff}}$$

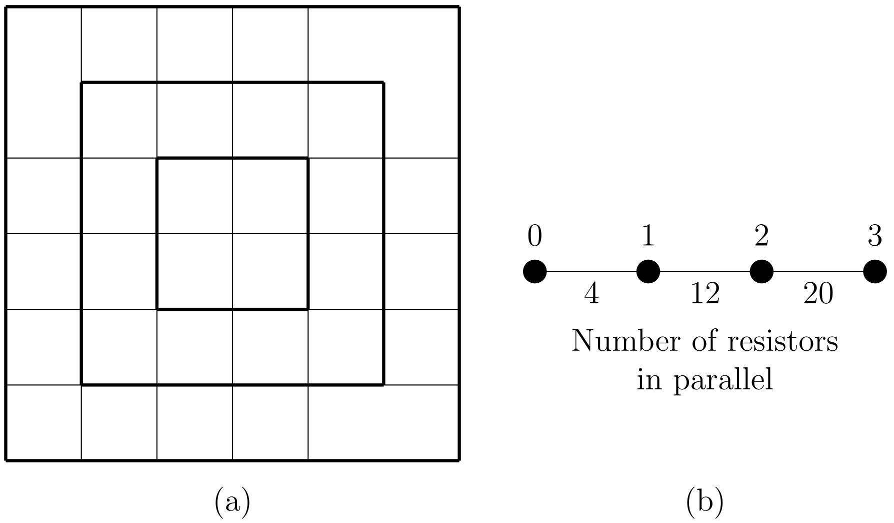


Figure 1.2: 2-dimensional lattice along with the linear network resulting from shorting resistors on the concentric squares about the origin.

In one dimension, the electrical network is just two series connections of n one ohm resistors connected in parallel. So, r_{eff} goes to infinity and the escape probability goes to zero as n goes to infinity. Thus, the walk in the unbounded one dimensional lattice will return to the origin with probability one.

Two dimensions

For the 2-dimensional lattice, consider a larger and larger square about the origin for the boundary as shown in Figure 1.2a and consider the limit of r_{eff} as the squares get larger. Shorting the resistors on each square can only reduce r_{eff}. Shorting the resistors results in the linear network shown in Figure 1.2b. As the paths get longer, the number of resistors in parallel also increases. So the resistor between node i and $i+1$ is really made up of $O(i)$ unit resistors in parallel. The effective resistance of $O(i)$ resistors in parallel is $1/O(i)$. Thus,

$$r_{eff} \geq \tfrac{1}{4} + \tfrac{1}{12} + \tfrac{1}{20} + \cdots = \tfrac{1}{4}(1 + \tfrac{1}{3} + \tfrac{1}{5} + \cdots) = \Theta(\ln n).$$

Since the lower bound on the effective resistance goes to infinity, the escape probability goes to zero for the 2-dimensional lattice.

Three dimensions

In three dimensions, the resistance along any path to infinity grows to infinity but the number of paths in parallel also grows to infinity. It turns out that r_{eff} remains finite and thus there is a nonzero escape probability.

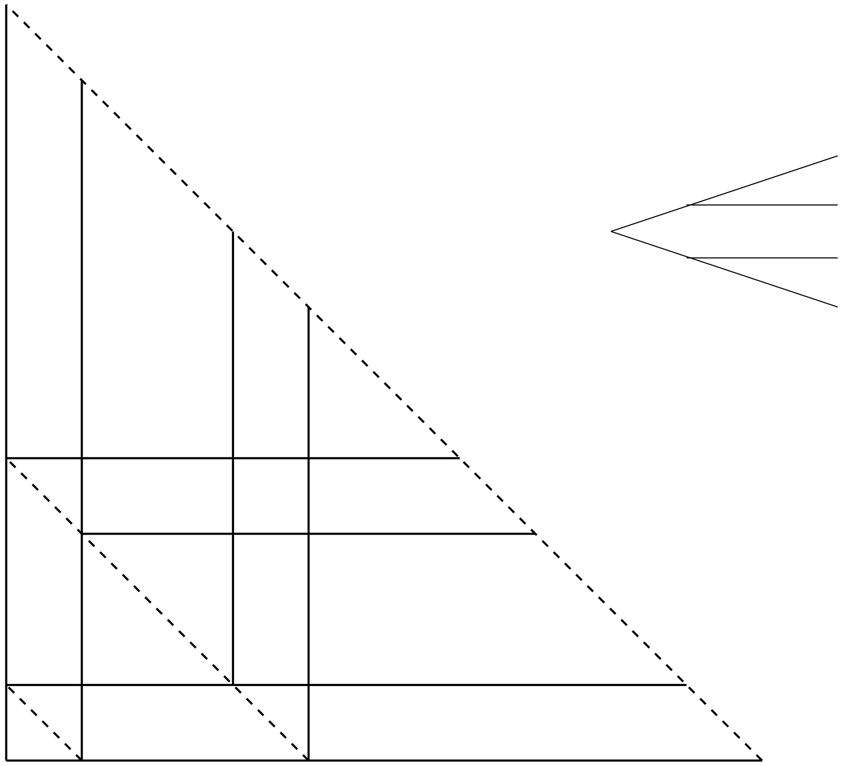


Figure 1.3: Paths in a 2-dimensional lattice obtained from the 3-dimensional construction applied in 2-dimensions.

The construction used in three dimensions is easier to explain first in two dimensions. Draw dotted diagonal lines at $x+y=2^n-1$. Consider two paths that start at the origin. One goes up and the other goes to the right. Each time a path encounters a dotted diagonal line, split the path into two, one which goes right and the other up. Where two paths cross, split the vertex into two, keeping the paths separate. By a symmetry argument, splitting the vertex does not change the resistance of the network. Remove all resistors except those on these paths. The resistance of the original network is less than that of the tree produced by this process since removing a resistor is equivalent to increasing its resistance to infinity.

The distances between splits increase and are 1, 2, 4, etc. At each split the number of paths in parallel doubles. Thus, the resistance to infinity in this two dimensional example is

$$\frac{1}{2}+\frac{1}{4}2+\frac{1}{8}4+\cdots=\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\cdots=\infty.$$

In the analogous three dimensional construction, paths go up, to the right, and out of the plane of the paper. The paths split three ways at planes given by $x+y+z=2^n-1$.

Each time the paths split the number of parallel segments triple. Segments of the paths between splits are of length 1, 2, 4, etc. and the resistance of the segments are equal to the lengths. The resistance out to infinity for the tree is

$$\tfrac{1}{3}+\tfrac{1}{9}2+\tfrac{1}{27}4+\cdots=\tfrac{1}{3}\left(1+\tfrac{2}{3}+\tfrac{4}{9}+\cdots\right)=\tfrac{1}{3}\tfrac{1}{1-\frac{2}{3}}=1$$

The resistance of the three dimensional lattice is less. Thus, in three dimensions the escape probability is nonzero. The upper bound on r_{eff} gives the lower bound

$$p_{escape}=\tfrac{1}{2d}\tfrac{1}{r_{eff}}\geq\tfrac{1}{6}.$$

A lower bound on r_{eff} gives an upper bound on p_{escape}. To get the upper bound on p_{escape}, short all resistors on surfaces of boxes at distances $1,2,3,$, etc. Then

$$r_{eff}\geq\tfrac{1}{6}\left[1+\tfrac{1}{9}+\tfrac{1}{25}+\cdots\right]\geq\tfrac{1.23}{6}\geq 0.2$$

This gives

$$p_{escape}=\tfrac{1}{2d}\tfrac{1}{r_{eff}}\geq\tfrac{5}{6}.$$

1.5 Random Walks on Directed Graphs

A major application of random walks on directed graphs comes from trying to establish the importance of pages on the World Wide Web. One way to do this would be to take a random walk on the web and rank pages according to their stationary probability. However, several situations occur in random walks on directed graphs that did not arise with undirected graphs. One difficulty occurs if there is a node with no out edges. In this case, the directed graph is not strongly connected and so Markov chain is not strongly connected either even though the underlying undirected graph may be connected. When the walk encounters this node the walk disappears. Another difficulty is that a node or a strongly connected component with no in edges is never reached. One way to resolve these difficulties is to introduce a random restart condition. At each step, with some probability r, jump to a node selected uniformly at random and with probability $1-r$ select an edge at random and follow it. If a node has no out edges, the value of r for that node is set to one. This has the effect of converting the graph to a strongly connected graph so that the stationary probabilities exist.

1.6 Finite Markov Processes

A *Markov process* is a random process in which the probability distribution for the future behavior depends only on the current state, not on how the process arrived at the current state. Markov processes are equivalent mathematically to random walks on directed graphs but the literature on the two topics developed separately with different terminology. Since much of the terminology of Markov processes appears in the literature on random walks, we introduce the terminology here to acquaint the reader with it.

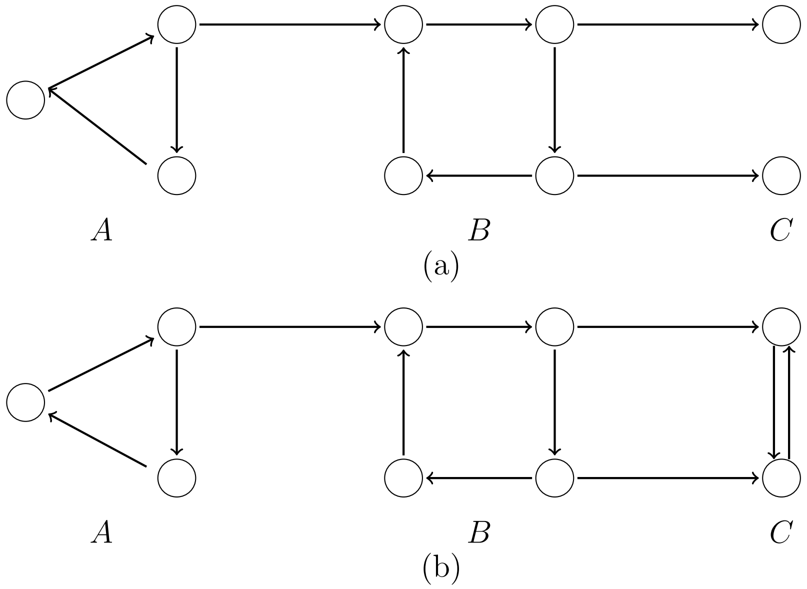


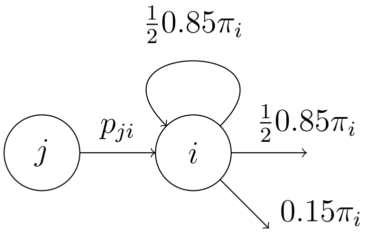
Figure 1.4: (a) A directed graph with nodes with no out out edges and a strongly connected component A with no in edges.
(b) A directed graph with three strongly connected components.

In a Markov process, nodes of the underlying graph are referred to as states. A state is *persistent* if it has the property that should the state ever be reached, the random process will return to it with probability one. This means that the state is in a strongly connected component with no out edges. Consider the directed graph in Figure 1.4b with three strongly connected components A, B, and C. Starting from any node in A there is a nonzero probability of eventually reaching any node in A. However, the probability of returning to a node in A is less than one and thus nodes in A and similarly nodes in B are not persistent. From any node in C, the walk will return with probability one to that node eventually since there is no way of leaving component C. Thus, nodes in C are persistent.

A state is *periodic* if it is contained only in cycles in which the greatest common divisor (gcd) of the cycle lengths is greater than one. A Markov process is *irreducible* if it consists of a single strongly connected component. An *ergodic* state is one that is aperiodic and persistent. A Markov process is *ergodic* if all states are ergodic. In graph theory this corresponds to a single strongly connected component that is aperiodic.

Page rank and hitting time

The page rank of a node in a directed graph is the stationary probability of the node. We assume some restart value, say $r = 0.15$, is used. The restart ensures that the graph is strongly connected. The page rank of a page is the fractional frequency with which the



$$\pi_i = \pi_j p_{ji} + \tfrac{0.85}{2}\pi_i$$

$$\pi_i = 1.74(\pi_j p_{ji})$$

Figure 1.5: Impact on page rank of adding a self loop

page will be visited over a long period of time. If the page rank is p, then the expected time between visits or return time is $1/p$. Notice that one can increase the pagerank of a page by reducing the return time and this can be done by creating short cycles.

Consider a node i with a single edge in from node j and a single edge out. The stationary probability $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi} P = \boldsymbol{\pi}$, and thus

$$\pi_i = \pi_j p_{ji}.$$

Adding a self-loop at i, results in a new equation

$$= pi_i = \pi_j p_{ji} + \frac{1}{2}\pi_i$$

or

$$\pi_i = 2\ \pi_j p_{ji}.$$

Of course, π_j would have changed too, but ignoring this for now, pagerank is doubled by the addition of a self-loop. Adding k self loops, results in the equation

$$\pi_i = \pi_j p_{ji} + \frac{k}{k+1}\pi_i,$$

and again ignoring the change in π_j, we now have $\pi_i = (k+1)\pi_j p_{ji}$. What prevents one from increasing the page rank of a page arbitrarily? The answer is the restart. We neglected the 0.15 probability that is taken off for the random restart. With the restart taken into account, the equation for π_i when there is no self-loop is

$$\pi_i = 0.85\pi_j p_{ji}$$

whereas, with k self-loops, the equation is

$$\pi_i = 0.85\pi_j p_{ji} + 0.85\frac{k}{k+1}\pi_i.$$

Adding a single loop only increases pagerank by a factor of 1.74 and adding k loops increases it by at most a factor of 6.67 for arbitrarily large k.

Hitting time

Related to page rank is a quantity called hitting time. Hitting time is closely related to return time and thus to the reciprocal of page rank. One way to return to a node v is by a path in the graph from v back to v. Another way is to start on a path that encounters a restart, followed by a path from the random restart node to v. The time to reach v after a restart is the hitting time. Thus, return time is clearly less than the expected time until a restart plus hitting time. The fastest one could return would be if there were only paths of length two since self loops are ignored in calculating page rank. If r is the restart value, then the loop would be traversed with at most probability $(1-r)^2$. With probability $r+(1-r)\,r=(2-r)\,r$ one restarts and then hits v. Thus, the return time is at least $(1-r)^2+(2-r)\,r\times$ (hitting time). Combining these two bounds yields

$$(1-r)^2+(2-r)\,rE\,(\text{hitting time})\leq E\,(\text{return time})\leq E\,(\text{hitting time})$$

The relationship between return time and hitting time can be used to see if a node has unusually high probability of short loops. However, there is no efficient way to compute hitting time for all nodes as there is for return time. For a single node v, one can compute hitting time by removing the edges out of the node v for which one is computing hitting time and then run the page rank algorithm for the new graph. The hitting time for v is the reciprocal of the page rank in the graph with the edges out of v removed. Since computing hitting time for each node requires removal of a different set of edges, the algorithm only gives the hitting time for one node at a time. Since one is probably only interested in the hitting time of nodes with low hitting time, an alternative would be to use a random walk to estimate the hitting time of low hitting time nodes.

Spam

Suppose one has a web page and would like to increase its page rank by creating some other web pages with pointers to the original page. The abstract problem is the following. We are given a directed graph G and a node v whose page rank we want to increase. We may add new nodes to the graph and add edges from v or from the new nodes to any nodes we want. We cannot add edges out of other nodes. We can also delete edges from v.

The page rank of v is the stationary probability for node v with random restarts. If we delete all existing edges out of v, create a new node u and edges (v,u) and (u,v), then the page rank will be increased since any time the random walk reaches v it will be captured in the loop $v\to u\to v$. A search engine can counter this strategy by more frequent random restarts.

A second method to increase page rank would be to create a star consisting of the node v at its center along with a large set of new nodes each with a directed edge to v. These new nodes will sometimes be chosen as the target of the random restart and hence the nodes increase the probability of the random walk reaching v. This second method is countered by reducing the frequency of random restarts.

Notice that the first technique of capturing the random walk increases page rank but does not effect hitting time. One can negate the impact of someone capturing the random walk on page rank by increasing the frequency of random restarts. The second technique of creating a star increases page rank due to random restarts and decreases hitting time. One can check if the page rank is high and hitting time is low in which case the page rank is likely to have been artificially inflated by the page capturing the walk with short cycles.

Personalized page rank

In computing page rank, one uses a restart probability, typically 0.15, in which at each step, instead of taking a step in the graph, the walk goes to a node selected uniformly at random. In personalized page rank, instead of selecting a node uniformly at random, one selects a node according to a personalized probability distribution. Often the distribution has probability one for a single node and whenever the walk restarts it restarts at that node.

Algorithm for computing personalized page rank

First, consider the normal page rank. Let α be the restart probability with which the random walk jumps to an arbitrary node. With probability $1-\alpha$ the random walk selects a node uniformly at random from the set of adjacent nodes. Let $\mathbf{p}$ be a row vector denoting the page rank and let G be the adjacency matrix with rows normalized to sum to one. Then

$$\mathbf{p} = \tfrac{\alpha}{n}\,(1,1,\ldots,1) + (1-\alpha)\,\mathbf{p}G$$

$$\mathbf{p}[I-(1-\alpha)G] = \frac{\alpha}{n}(1,1,\ldots,1)$$

or

$$\mathbf{p} = \tfrac{\alpha}{n}\,(1,1,\ldots,1)\,[I-(1-\alpha)\,G]^{-1}.$$

Thus, in principle, $\mathbf{p}$ can be found by computing the inverse of $[I-(1-\alpha)G]^{-1}$. But this is far from practical since for the whole web one would be dealing with matrices with billions of rows and columns. A more practical procedure is to run the random walk and observe using the basics of the power method in Chapter **??** that the process converges to the solution $\mathbf{p}$.

For the personalized page rank, instead of restarting at an arbitrary vertex, the walk restarts at a designated vertex. More generally, it may restart in some specified neighborhood. Suppose the restart selects a vertex using the probability distribution s. Then, in the above calculation replace the vector $\frac{1}{n}(1,1,\ldots,1)$ by the vector $\mathbf{s}$. Again, the computation could be done by a random walk. But, we wish to do the random walk calculation for personalized pagerank quickly since it is to be performed repeatedly. With more care this can be done, though we do not describe it here.

1.7 Markov Chain Monte Carlo

The Markov Chain Monte Carlo method is a technique for sampling a multivariate probability distribution $p(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ is the set of variables. Given the probability distribution $p(\mathbf{x})$, one might wish to calculate the marginal distribution

$$p(x_1) = \sum_{x_2,\ldots,x_d} p(x_1, \ldots, x_d)$$

or the expectation of some function$f(\mathbf{x})$

$$E(f) = \sum_{x_1,\ldots,x_d} f(x_1, \ldots, x_d)\, p(x_1, \ldots, x_d).$$

The difficulty is that both computations require a summation over an exponential number of values. If each x_i can take on a value from the set $\{1, 2, \ldots, n\}$ of values, then there are n^d possible values for $\mathbf{x}$. One could compute an approximate answer by generating a sample set of values for $\mathbf{x} = (x_1, \ldots, x_d)$ according to the distribution $p(x_1, \ldots, x_d)$. This is done by designing a Markov chain whose stationary probabilities are exactly $p(x_1, x_2, \ldots, x_d)$ and running the chain for a sufficiently large number of steps and averaging f over the states seen in the run. The number of steps must be large enough that we are close to the limit which is the stationary distribution. In the rest of this section, we will show that under some mild conditions, the number of steps needed grows only polynomially, though the total number of states grows exponentially with d.

For ease of explanation, assume that the variables take on values from some finite set. Create a directed graph with one node corresponding to each possible value of $\mathbf{x}$. A random walk on the the graph is designed so that the stationary probability of the walk is $p(\mathbf{x})$. The walk is designed by specifying the probability of the transition from one node to another in such a way as to achieve the desired stationary distribution. Two common techniques for designing the walks are the Metropolis-Hasting algorithm and Gibbs sampling. We will see that the sequence of nodes after a sufficient number of steps of the walk provides a good sample of the distribution. The number of steps the walk needs to take depends on its convergence rate to its stationary distribution. We will show that this rate is related to a natural quantity called the minimum escape probability (MEP).

We used $\mathbf{x} \in \mathbf{R}^d$ to emphasize that our distributions are multi-variate. From a Markov chain perspective, each value $\mathbf{x}$ can take on is a state, i.e., a node of the graph on which the random walk takes place. Henceforth, we will use the subscripts $i, j, k, \ldots$ to denote states and will use p_i instead of $p(x_1, x_2, \ldots, x_d)$ to denote the probability of the state corresponding to a given set of values for the variables. Recall that in the Markov chain terminology, nodes of the graph are called states.

Recall the notation that $\mathbf{p^{(t)}}$ is the row vector of probabilities of the random walk being at each state (node of the graph) at time t. So, $\mathbf{p^{(t)}}$ has as many components as there are states and its i^{th} component, $p_i^{(t)}$, is the probability of being in state i at time t. Recall the long-term (t-step) average is

$$\mathbf{a^{(t)}} = \frac{1}{t}\left[\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right]. \tag{1.3}$$

The expected value of the function f under the probability distribution p is $E(f) = \sum_i f_i p_i$. Our estimate of this quantity will be the average value of f at the states seen in a t step run. Call this estimate a. Clearly, the expected value of a is

$$E(a) = \sum_i f_i a_i^{(t)}.$$

The expectation here is with respect to the "coin tosses" of the algorithm, not with respect to the underlying distribution p. Letting $f_{\max}$ denote the maximum absolute value of f. It is easy to see that

$$\left| E(a) - \sum_i f_i p_i \right| \leq f_{\max} \sum_i |p_i - a_i^{(t)}| = f_{\max} |\mathbf{p} - \mathbf{a^{(t)}}|_1 \tag{1.4}$$

where the quantity $|\mathbf{p} - \mathbf{a^{(t)}}|_1$ is the l_1 distance between the probability distributions $\mathbf{p}$ and $\mathbf{a^{(t)}}$ and is often called the "total variation distance" between the distributions. We will build tools to upper bound $|\mathbf{p} - \mathbf{a^{(t)}}|_1$. Since $\mathbf{p}$ is the steady state distribution, the t for which $|\mathbf{p} - \mathbf{a^{(t)}}|_1$ becomes small is determined by the rate of convergence of the Markov chain to its steady state.

The following proposition is often useful.

Proposition 1.9 *For two probability distributions* $\mathbf{p}$ *and* $\mathbf{q}$, $|\mathbf{p} - \mathbf{q}|_1 = 2\sum_i (p_i - q_i)^+ = 2\sum_i (q_i - p_i)^+$.

The proof is left as an exercise (Exercise 1.34).

1.7.1 Time Reversibility

Definition: A Markov chain is said to be *time-reversible* if for the steady state probabilities $\boldsymbol{\pi}$, $\pi_i p_{ij} = \pi_j p_{ji}$ for all i and j. ■

The phrase "time-reversible" comes from the following fact. For a time-reversible Markov chain started in the steady state, the probability of a path (sequence of states) is the same as its reversal. That is

$$\pi_{i_1}\ p_{i_1,i_2}p_{i_2,i_3}\cdots p_{i_{k-1},i_k}=\pi_{i_k}\ p_{i_k,i_{k-1}}p_{i_{k-1},i_{k-2}}\cdots p_{i_2,i_1}.$$

Given only the sequence of states seen, one cannot tell if time runs forward or backward, both having the same probability. More important is the fact that time reversibility simplifies the underlying mathematics as illustrated in the following lemma. The lemma states that if a probability distribution $\mathbf{q}$ has the property that the probability of traversing each edge is the same in both directions, then the probability distribution must be the steady state distribution of the Markov chain. The lemma is used frequently.

Lemma 1.10 *In a strongly connected Markov chain with transition probabilities p_{ij}, if a vector $\mathbf{q}$ with non-negative components satisfies*

$$q_ip_{ij}=q_jp_{ji}$$

for all i and j, then $q_i/\sum\limits_k q_k$ is the stationary probability of node i.

Proof: Since the chain is strongly connected, there is a unique stationary probability vector $\boldsymbol{\pi}$ satisfying the equations $\boldsymbol{\pi}P=\boldsymbol{\pi}$ and $\sum_i \pi_i=1$. Now $\mathbf{q}/\sum_k q_k$ satisfies these equations since $\sum_i q_i/\sum_k q_k=1$ and for each fixed j, $\sum_i q_ip_{ij}=\sum_i q_jp_{ji}=q_j\sum_i p_{ji}=q_j$. Thus $\mathbf{q}$ must be the steady state distribution. ∎

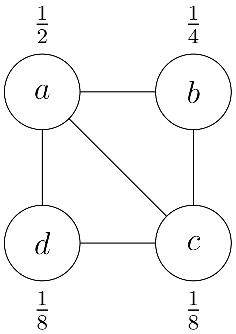
1.7.2 Metropolis-Hasting Algorithm

Metropolis-Hasting algorithm is a general method to design a Markov chain whose stationary distribution is a given target distribution p. Start with a connected undirected graph G on the set of states. For example, if the states are the lattice points $(x_1,x_2,\ldots,x_d)$ in $\mathbf{R}^d$ with $\{x_i\in\{0,1,2,,\ldots,n\}\}$, then G is the lattice graph with $2d$ coordinate edges at each interior vertex. In general, let r be the maximum degree of any node of G. The transitions of the Markov chain are defined as follows. At state i select neighbor j with probability $\frac{1}{r}$. Since the degree of i may be less than r, with some probability no edge is selected and the walk remains at i. If a neighbor j is selected and $p_j\geq p_i$, go to j. If $p_j<p_i$, go to j with probability p_j/p_i and stay at i with probability $1-\frac{p_j}{p_i}$. Intuitively, this favors "heavier" states with higher p values. For i and j adjacent in G, $p_{ij}=\frac{1}{r}\min\left(1,\frac{p_j}{p_i}\right)$ and $p_{ii}=1-\sum\limits_{j\neq i}p_{ij}$. Then

$$p_ip_{ij}=\frac{p_i}{r}\min\left(1,\frac{p_j}{p_i}\right)=\frac{1}{r}\min(p_i,p_j)=\frac{p_j}{d}\min\left(1,\frac{p_i}{p_j}\right)=p_jp_{ji}.$$

By Lemma 1.10, the stationary probabilities are $p(\mathbf{x})$ as desired.

$p(a) = \frac{1}{2}$
$p(b) = \frac{1}{4}$
$p(c) = \frac{1}{8}$
$p(d) = \frac{1}{8}$



$a \to b$	$\frac{1}{3}\frac{1}{4}\frac{2}{1} = \frac{1}{6}$	$c \to a$	$\frac{1}{3}$
$a \to c$	$\frac{1}{3}\frac{1}{8}\frac{2}{1} = \frac{1}{12}$	$c \to b$	$\frac{1}{3}$
$a \to d$	$\frac{1}{3}\frac{1}{8}\frac{2}{1} = \frac{1}{12}$	$c \to d$	$\frac{1}{3}$
$a \to a$	$\frac{2}{3}$	$c \to c$	$1 - \frac{1}{3} - \frac{1}{3} - \frac{1}{3} = 0$
$b \to a$	$\frac{1}{3}$	$d \to a$	$\frac{1}{3}$
$b \to c$	$\frac{1}{3}\frac{1}{8}\frac{4}{1} = \frac{1}{6}$	$d \to c$	$\frac{1}{3}$
$b \to b$	$1 - \frac{1}{3} - \frac{1}{6} = \frac{1}{2}$	$d \to d$	$1 - \frac{1}{3} - \frac{1}{3} = \frac{1}{3}$

$$\begin{aligned} p(a) &= p(a)p(a \to a) + p(b)p(b \to a) + p(c)p(c \to a) + P(d)p(d \to a) \\ &= \tfrac{1}{2}\,\tfrac{2}{3} + \tfrac{1}{4}\,\tfrac{1}{3} + \tfrac{1}{8}\,\tfrac{1}{3} + \tfrac{1}{8}\,\tfrac{1}{3} = \tfrac{1}{2} \end{aligned}$$

$$\begin{aligned} p(b) &= p(a)p(a \to b) + p(b)p(b \to b) + p(c)p(c \to b) \\ &= \tfrac{1}{2}\,\tfrac{1}{6} + \tfrac{1}{4}\,\tfrac{1}{2} + \tfrac{1}{8}\,\tfrac{1}{3} = \tfrac{1}{4} \end{aligned}$$

$$\begin{aligned} p(c) &= p(a)p(a \to c) + p(b)p(b \to c) + p(c)p(c \to c) + P(d)p(d \to c) \\ &= \tfrac{1}{2}\,\tfrac{1}{12} + \tfrac{1}{4}\,\tfrac{1}{6} + \tfrac{1}{8}\,0 + \tfrac{1}{8}\,\tfrac{1}{3} = \tfrac{1}{8} \end{aligned}$$

$$\begin{aligned} p(d) &= p(a)p(a \to d) + p(c)p(c \to d) + P(d)p(d \to d) \\ &= \tfrac{1}{2}\,\tfrac{1}{12} + \tfrac{1}{8}\,\tfrac{1}{3} + \tfrac{1}{8}\,\tfrac{1}{3} = \tfrac{1}{8} \end{aligned}$$

Figure 1.6: Using the Metropolis-Hasting algorithm to set probabilities for a random walk so that the stationary probability will be a desired probability.

Example: Consider the graph in Figure 1.6. Using the Metropolis-Hasting algorithm, assign transition probabilities so that the stationary probability of a random walk is $p(a) = \frac{1}{2}$, $p(b) = \frac{1}{4}$, $p(c) = \frac{1}{8}$, and $p(d) = \frac{1}{8}$. The maximum degree of any vertex is three so at a the probability of taking the edge (a, b) is $\frac{1}{3}\frac{1}{4}\frac{2}{1}$ or $\frac{1}{6}$. The probability of taking the edge (a, c) is $\frac{1}{3}\frac{1}{8}\frac{2}{1}$ or $\frac{1}{12}$ and of taking the edge (a, d) is $\frac{1}{3}\frac{1}{8}\frac{2}{1}$ or $\frac{1}{12}$. Thus the probability of staying at a is $\frac{2}{3}$. The probability of taking the edge from b to a is $\frac{1}{3}$. The probability of taking the edge from c to a is $\frac{1}{3}$ and the probability of taking the edge from d to a is $\frac{1}{3}$. Thus the stationary probability of a is $\frac{1}{4}\frac{1}{3} + \frac{1}{8}\frac{1}{3} + \frac{1}{8}\frac{1}{3} + \frac{1}{2}\frac{2}{3} = \frac{1}{2}$, which is what is desired. ■

1.7.3 Gibbs Sampling

Gibbs sampling is another Markov Chain Monte Carlo method to sample from a multivariate probability distribution. Let $p(\mathbf{x})$ be the target distribution where $\mathbf{x} = (x_1, \ldots, x_d)$. Gibbs sampling consists of a random walk on a graph whose vertices correspond to the values of $\mathbf{x} = (x_1, \ldots, x_d)$ and in which there is an edge from $\mathbf{x}$ to $\mathbf{y}$ if $\mathbf{x}$ and $\mathbf{y}$ differ in only one coordinate.

To generate samples of $\mathbf{x} = (x_1, \ldots, x_d)$ with a target distribution $p\left(\mathbf{x}\right)$, the Gibbs sampling algorithm repeats the following steps. One of the variables x_i is chosen to be updated. Its new value is chosen based on the marginal probability of x_i with the other variables fixed. There are two commonly used schemes to determine which x_i to update. One scheme is to choose x_i randomly, the other is to choose x_i by sequentially scanning from x_1 to x_d.

Suppose that $\mathbf{x}$ and $\mathbf{y}$ are two states that differ in only one coordinate x_i. Then, in the scheme where a coordinate is randomly chosen to modify, the probability $p_{\mathbf{xy}}$ of going from $\mathbf{x}$ to $\mathbf{y}$ is

$$p_{\mathbf{xy}} = \frac{1}{d} p(y_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d).$$

The normalizing constant is $1/d$ since for a given value i the probability distribution of $p(y_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ sums to one, and thus summing i over the d-dimensions results in a value of d. Similarly,

$$p_{\mathbf{yx}} = \frac{1}{d} p(x_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d).$$

Here use was made of the fact that for $j \neq i$, $x_j = y_j$.

It is simple to see that this chain is time reversible with stationary probability proportional to $p\left(\mathbf{x}\right)$. Rewrite $p_{\mathbf{xy}}$ as

$$\begin{aligned} p_{\mathbf{xy}} &= \frac{1}{d} \frac{p(y_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d) p(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)}{p(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_d)}{p(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(\mathbf{y})}{p(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)} \end{aligned}$$

again using $x_j = y_j$ for $j \neq i$. Similarly write

$$p_{\mathbf{yx}} = \frac{1}{d} \frac{p(\mathbf{x})}{p(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)}$$

from which it follows that $p(\mathbf{x}) p_{xy} = p(\mathbf{y}) p_{yx}$. By Lemma 1.10 the stationary probability of the random walk is $p(\mathbf{x})$.

1.8 Convergence to Steady State

The Metropolis-Hasting algorithm and Gibbs sampling both involve a random walk. Initial states of the walk are highly dependent on the start state of the walk. An important question is how fast does the walk start to reflect the stationary probability of the

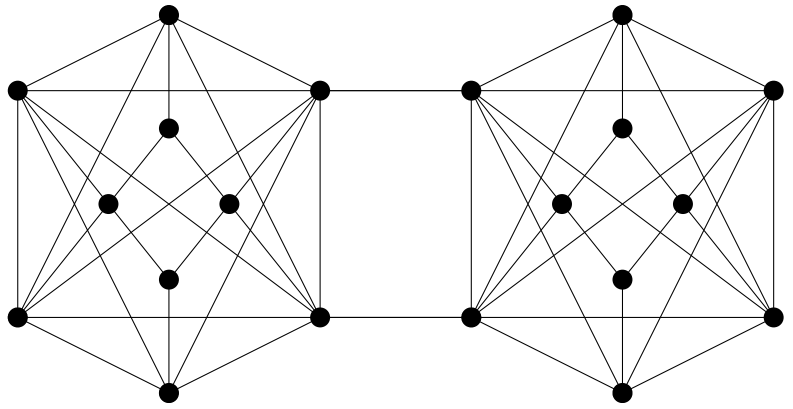


Figure 1.7: A constriction.

Markov process. If the convergence time was proportional to the number of states the algorithms would not be very useful, since as we remarked, the number of states can be exponentially large.

There are clear examples of connected chains that take a long time to converge. A chain with a constriction (see Figure 1.7) takes a long time to converge since the walk is unlikely to reach the narrow passage between the two halves, which are both reasonably big. The interesting thing is that a converse is also true. If there is no constriction, then the chain converges fast. We show this shortly.

A function is unimodal if it has a single maximum, i.e., it increases and then decreases.. A unimodal function like the normal density has no constriction blocking a random walk from getting out of a large set, whereas a bimodal function can have a constriction. Interestingly, many common multivariate distributions as well as univariate probability distributions like the normal and exponential are unimodal and sampling according to these distributions can be done using the methods here.

A natural problem is estimating the probability of a convex region in d-space according to a normal distribution. Let R be the region defined by the inequality $x_1+x_2+\cdots+x_{d/2} \leq x_{(d/2)+1}+\cdots+x_d$. Pick a sample according to the normal distribution and accept the sample if it satisfies the inequality. If not, reject the sample and retry until one gets a number of samples satisfying the inequality. Then the probability of the region is approximated by the fraction of the samples that satisfied the inequality. However, suppose R was the region $x_1 + x_2 + \cdots + x_{d-1} \leq x_d$. The probability of this region is exponentially small in d and so rejection sampling runs into the problem that we need to pick exponentially many samples before we expect to accept even one sample. This second situation is typical. Imagine computing the probability of failure of a system. The object of design is to

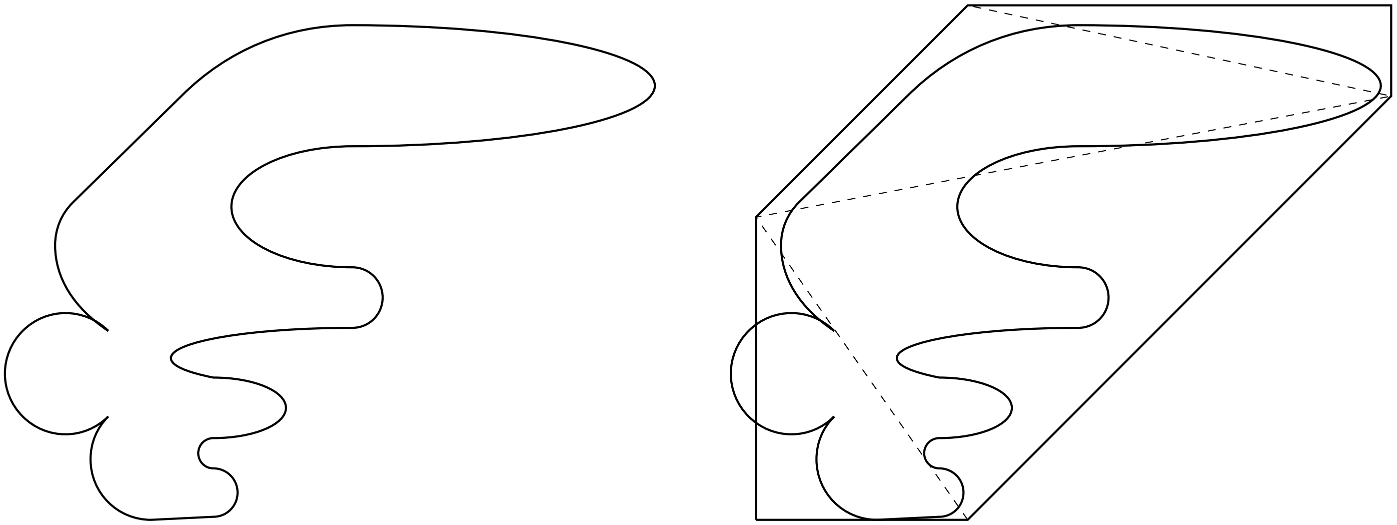


Figure 1.8: Area enclosed by curve.

make the system reliable, so the failure probability is likely to be very low and rejection sampling will take a long time to estimate the failure probability.

A similar problem is one of computing areas and volumes. First consider the problem of computing the area enclosed by the curve in Figure 1.8. One method would be to find a "nicer" enclosing shape. The picture on the right shows a convex polygon whose area we can compute in closed form by adding up the areas of the triangles. Throw darts at the larger shape, i.e., pick samples uniformly at random from the larger shape, and estimate the ratio of areas by the proportion of samples that land in the area enclosed by the curve.

Such methods fail in higher dimensions. For example, to compute the volume of a d-dimensional sphere by enclosing the sphere in a cube where the ratio of volume of the sphere to the cube is exponentially small, requires throwing exponentially many darts before getting any nonzero answer.

A different way to solve the problem of drawing a uniform random sample from a d-dimensional region is to put a grid on the region and do a random walk on the grid points. At each time, pick one of the $2d$ coordinate neighbors of the current grid point, each with probability $1/(2d)$, then go to the neighbor if it is still in the set; otherwise, stay put and repeat. This can be shown to lead to a polynomial time algorithm for drawing a uniform random sample from a bounded convex d-dimensional region. It turns out that this can be used to estimate volumes of such a region by immersing the region in a magnified copy of itself intersected with a nice object like a cube. We do not give the details here.

In general, there could be constrictions that prevent rapid convergence to the stationary probability. However, if the set is convex in any number of dimensions, then there are

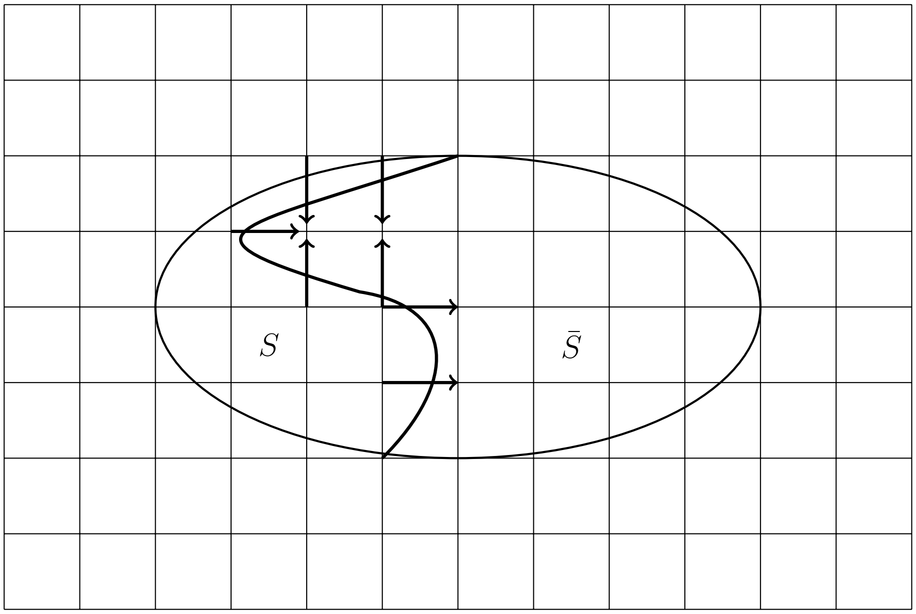


Figure 1.9: Each grid point in the ellipse is a state. The set of states in the ellipse is divided into two sets, S and $\bar{S}$, by the curve. The transitions from S to $\bar{S}$, which contribute to $\Phi(S)$, are marked with arrows.

no constrictions and there is rapid convergence although the proof of this is beyond the scope of this book.

Suppose $\mathbf{q}$ is any probability distribution on the states. Execute one step of the Markov chain starting with distribution $\mathbf{q}$. Then the amount of probability that "flows" from i to j is $q_i p_{ji}$. If S and T are two possibly intersecting subsets of states, the total flow from S to T is $\sum\limits_{i \in S, j \in T} q_i p_{ij}$. We use the notation

$$\text{flow}(i,j) = q_i p_{ij}$$

and

$$\text{flow}(S,T)) = \sum_{i \in S, j \in T} q_i p_{ij}.$$

We define below a combinatorial measure of constriction for a Markov chain, called the *minimum escape probability*, and relate this quantity to the rate of convergence to the stationarity probability.[1]

[1] In the Markov Chain literature, the word "conductance" is often used for minimum escape probability. Here, we have reserved the word conductance for the natural electrical quantity which is the reciprocal of resistance.

Definition: For a subset S of states of a Markov chain with stationary probabilities $\boldsymbol{\pi}$, define $\Phi(S)$, the *escape probability of* S, by

$$\Phi(S) = \frac{\text{flow}(S, \bar{S})}{\boldsymbol{\pi}(S)}.$$

The escape probability of S is the probability of taking a step from S to outside S conditioned on starting in S where the stationary probability at state i in S is proportional to its stationary probability, i.e., $\pi_i/\boldsymbol{\pi}(S)$. The *minimum escape probability MEP* of the Markov chain, denoted Φ, is defined by

$$\Phi = \min_{\substack{S \\ \boldsymbol{\pi}(S) \leq 1/2}} \Phi(S).$$

■

The restriction to sets with $\boldsymbol{\pi} \leq 1/2$ in the definition of ϕ is natural. One does not need to escape from big sets. Note that a constriction would mean a small Φ.

Definition: Fix an $\varepsilon > 0$. The *ε-mixing time* of a Markov chain is the minimum integer t such that for any starting distribution $\mathbf{p^{(0)}}$, the 1-norm distance between the t-step running average probability distribution $\mathbf{a^{(t)}}$ and the stationary distribution is at most ε. ■

The theorem below states that if the minimum escape probability Φ is large, then there is fast convergence of the running average probability. Intuitively, if Φ is large then the walk rapidly leaves any subset of states. Later we will see examples where the mixing time is much smaller than the cover time. That is, the number of steps before a random walk reaches a random state independent of its starting state is much smaller than the average number of steps needed to reach every state. We assume time reversibility, namely that $\pi_i p_{ij} = \pi_j p_{ji}$.

Theorem 1.11 *The ε mixing time of a time-reversible Markov chain is*

$$O\left(\frac{\ln(1/\pi_{min})}{\Phi^2 \varepsilon^3}\right).$$

Proof: Recall that $\mathbf{a^{(t)}}$ is the long term average probability distribution. Let

$$t = \frac{c \ln(1/\pi_{\min})}{\Phi^2 \varepsilon^2},$$

for a suitable constant c. For convenience, let $\mathbf{a} = \mathbf{a^{(t)}}$. We need to show that $|\mathbf{a} - \boldsymbol{\pi}| \leq \varepsilon$.

Let v_i denote the ratio of the long term average probability at time t divided by the stationary probability. Thus $v_i = \frac{a_i}{\pi_i}$. Renumber states so that $v_1 \leq v_2 \leq \cdots$. Since

$\mathbf{a}P$ is the probability vector after executing one step of the Markov chain starting with probabilities $\mathbf{a}$, $\mathbf{a}-\mathbf{a}P$ is the net loss of probability due to the step. Let k be any integer with $v_k > 1$. Let $A = \{1, 2, \ldots, k\}$. The net loss of probability from the set A in one step is $\sum_{i=1}^{k}(a_i - (\mathbf{a}P)_i) \leq \frac{2}{t}$ as in the proof of Theorem 1.1.

Another way to reckon the net loss of probability from A is to take the difference of the probability flow from A to $\bar{A}$ and the flow from $\bar{A}$ to A. By time-reversibility, for $i < j$,

$$\text{flow}(i,j) - \text{flow}(j,i) = \pi_i p_{ij} v_i - \pi_j p_{ji} v_j = \pi_j p_{ji}(v_i - v_j) \geq 0,$$

Thus for any $l \geq k$, the flow from A to $\{k+1, k+2, \ldots, l\}$ minus the flow from $\{k+1, k+2, \ldots, l\}$ is non-negative. The net loss from A is at least

$$\sum_{\substack{i\leq k \\ j>l}} \pi_j p_{ji}(v_i - v_j) \geq (v_k - v_{l+1}) \sum_{\substack{i\leq k \\ j>l}} \pi_j p_{ji}.$$

Thus,

$$(v_k - v_{l+1}) \sum_{\substack{i\leq k \\ j>l}} \pi_j p_{ji} \leq \frac{2}{t}.$$

If $\pi(\{i|v_i \leq 1\}) \leq \varepsilon/2$, then

$$|\mathbf{a} - \boldsymbol{\pi}|_1 = 2 \sum_{\substack{i \\ v_i\leq 1}} (1 - v_i)\pi_i \leq \varepsilon,$$

so we are done. Assume $\pi(\{i|v_i \leq 1\}) > \varepsilon/2$ so that $\pi(A) \geq \varepsilon \min(\pi(A), \pi(\bar{A}))/2$. Choose l to be the largest integer greater than or equal to k so that $\sum_{j=k+1}^{l} \pi_j \leq \varepsilon\Phi\pi(A)/2$. Since

$$\sum_{i=1}^{k} \sum_{j=k+1}^{l} \pi_j p_{ji} \leq \sum_{j=k+1}^{l} \pi_j \leq \varepsilon\Phi\pi(A)/2$$

by the definition of MEP,

$$\sum_{i\leq k<j} \pi_j p_{ji} \geq \Phi \min(\pi(A), \pi(\bar{A})) \geq \varepsilon\Phi\pi(A).$$

Thus $\sum_{\substack{i\leq k \\ j>l}} \pi_j p_{ji} \geq \varepsilon\Phi\pi(A)/2$ and substituting into the above inequality gives

$$v_k - v_{l+1} \leq \frac{8}{t\varepsilon\Phi\pi(A)}. \tag{1.5}$$

Now, divide $\{1, 2, \ldots\}$ into groups as follows. The first group G_1 is $\{1\}$. In general, if the r^{th} group G_r begins with state k, the next group G_{r+1} begins with state $l+1$ where l is as defined above. Let i_0 be the largest integer with $v_{i_0} > 1$. Stop with G_m, if G_{m+1} would begin with an $i > i_0$. If group G_r begins in i, define $u_r = v_i$. Let $\rho = 1 + \frac{\varepsilon\Phi}{2}$.

$$|\mathbf{a} - \boldsymbol{\pi}|_1 \leq 2\sum_{i=1}^{i_0} \pi_i(v_i - 1) \leq \sum_{r=1}^{m} \pi(G_r)(u_r - 1) = \sum_{r=1}^{m} \pi(G_1 \cup G_2 \cup \ldots \cup G_r)(u_r - u_{r+1}),$$

where the analog of integration by parts (for sums) is used in the last step and used the convention that $u_{m+1} = 1$. Since $u_r - u_{r+1} \leq 8/\varepsilon\Phi\pi(G_1 \cup \ldots \cup G_r)$, the sum is at most $8m/t\varepsilon\Phi$. Since $\pi_1 + \pi_2 + \cdots + \pi_{l+1} \geq \rho(\pi_1 + \pi_2 + \cdots + \pi_k)$,

$$m \leq \ln_\rho(1/\pi_1) \leq \ln(1/\pi_1)/(\rho - 1).$$

Thus $|\mathbf{a} - \boldsymbol{\pi}|_1 \leq O(\ln(1/\pi_{\min})/t\Phi^2\varepsilon^2) \leq \varepsilon$ for a suitable choice of c and this completes the proof. ■

1.8.1 Using Minimum Escape Probability to Prove Convergence

We now give some examples where Theorem 1.11 is used to bound the minimum escape probability and hence show rapid convergence. For the first example, consider a random walk on an undirected graph consisting of an n-vertex path with self-loops at the both ends. With the self loops, the stationary probability is a uniform $\frac{1}{n}$ over all vertices. The set with minimum escape probability consists of the first $n/2$ vertices, for which $\text{flow}(S, \bar{S}) = \pi_{n/2}p_{n/2,1+n/2} = \Omega(\frac{1}{n})$ and $\pi(S) = \frac{1}{2}$. Thus

$$\Phi(S) = \frac{\text{flow}(S, \bar{S})}{\boldsymbol{\pi}(S)} = 2\pi_{\frac{n}{2}}\, p_{\frac{n}{2},\frac{n}{2}+1} = \Omega(1/n).$$

By Theorem 1.11, for ε a constant such as $1/100$, after $O(n^2 \log n)$ steps, $|\mathbf{a}^{(\mathbf{t})} - \boldsymbol{\pi}|_1 \leq 1/100$. For this graph, the hitting time and the cover time are $O(n^2)$. In many interesting cases, the mixing time may be much smaller than the cover time. We will see such an example later.

For the second example, consider the $n \times n$ lattice in the plane where from each point there is a transition to each of the coordinate neighbors with probability ¼. At the boundary there are self-loops with probability 1-(number of neighbors)/4. It is easy to see that the chain is connected. Since $p_{ij} = p_{ji}$, the function $f_i = 1/n^2$ satisfies $f_ip_{ij} = f_jp_{ji}$ and by Lemma 1.10 is the stationary probability. Consider any subset S consisting of at most half the states. For at least half the states (x, y) in S, (each state is indexed by its x and y coordinate), either row x or column y intersects $\bar{S}$ (Exercise 1.35). Each state in S adjacent to a state in $\bar{S}$ contributes $\Omega(1/n^2)$ to the $\text{flow}(S, \bar{S})$. Thus,

$$\text{flow}(S, \bar{S}) = \sum_{i \in S}\sum_{j \notin S} \pi_i p_{ij} \geq \frac{\boldsymbol{\pi}(S)}{2}\frac{1}{n^2}$$

establishing that

$$\Phi = \frac{\text{flow}(S, \bar{S})}{\boldsymbol{\pi}(S)} \geq \frac{1}{2n}.$$

By Theorem 1.11, after $O(n^2 \ln n/\varepsilon^2)$ steps, $|\mathbf{a^{(t)}} - \boldsymbol{\pi}|_1 \leq 1/100$.

Next consider the $n \times n \times n \cdots \times n$ grid in d-dimensions with a self-loop at each boundary point with probability $1 -$ (number of neighbors)$/2d$. The self loops make all π_i equal to n^{-d}. View the grid as an undirected graph and consider the random walk on this undirected graph. Since there are n^d states, the cover time is at least n^d and thus exponentially dependent on d. It is possible to show (Exercise 1.49) that MEP is $\Omega(1/dn)$. Since all π_i are equal to n^{-d}, the mixing time is $O(d^3 n^2 \ln n/\varepsilon^2)$, which is polynomially bounded in n and d.

Next consider a random walk on a connected n vertex undirected graph where at each vertex all edges are equally likely. The stationary probability of a vertex equals the degree of the vertex divided by the sum of degrees which equals twice the number of edges. The sum of the vertex degrees is at most n^2 and thus, the steady state probability of each vertex is at least $\frac{1}{n^2}$. Since the degree of a vertex is at most n, the probability of each edge at a vertex is at least $\frac{1}{n}$. For any S,

$$\text{flow}(S, \bar{S}) \geq \frac{1}{n^2}\frac{1}{n} = \frac{1}{n^3}.$$

Thus the minimum escape probability is at least $\frac{1}{n^3}$. Since $\pi_{\min} \geq \frac{1}{n^2}$, $\ln \frac{1}{\pi_{min}} = O(\ln n)$. Thus, the mixing time is $O(n^6 (\ln n)/\varepsilon^2)$.

For our final example, consider the interval $[-1, 1]$. Let δ be a "grid size" specified later and let G be the graph consisting of a path on the $\frac{2}{\delta} + 1$ vertices $\{-1, -1 + \delta, -1 + 2\delta, \ldots, 1 - \delta, 1\}$ having self loops at the two ends. Let $\pi_x = ce^{-\alpha x^2}$ for $x \in \{-1, -1+\delta, -1+2\delta, \ldots, 1-\delta, 1\}$ where $\alpha > 1$ and c has been adjusted so that $\sum_x \pi_x = 1$.

We now describe a simple Markov chain with the π_x as its stationary probability and argue its fast convergence. With the Metropolis-Hastings' construction, the transition probabilities are

$$p_{x,x+\delta} = \frac{1}{2} \min\left(1, \frac{e^{-\alpha(x+\delta)^2}}{e^{-\alpha x^2}}\right) \text{ and } p_{x,x-\delta} = \frac{1}{2} \min\left(1, \frac{e^{-\alpha(x-\delta)^2}}{e^{-\alpha x^2}}\right).$$

Let S be any subset of states with $\boldsymbol{\pi}(S) \leq \frac{1}{2}$. Consider the case when S is an interval

$[k\delta, 1]$ for $k \geq 1$. It is easy to see that

$$\begin{aligned}
\boldsymbol{\pi}(S) &\leq \int_{x=(k-1)\delta}^{\infty} ce^{-\alpha x^2}\,dx \\
&\leq \int_{(k-1)\delta}^{\infty} \frac{x}{(k-1)\delta} ce^{-\alpha x^2}\,dx \\
&= O\left(\frac{ce^{-\alpha((k-1)\delta)^2}}{\alpha(k-1)\delta}\right).
\end{aligned}$$

Now there is only one edge from S to $\bar{S}$ and

$$\text{flow}(S, \bar{S}) = \sum_{i \in S} \sum_{j \notin S} \pi_i p_{ij} = \pi_{k\delta} p_{k\delta,(k-1)\delta} = \min(ce^{-\alpha k^2\delta^2}, ce^{-\alpha(k-1)^2\delta^2}) = ce^{-\alpha k^2\delta^2}.$$

Using $1 \leq k \leq 1/\delta$ and $\alpha \geq 1$, the minimum escape probability of S is

$$\begin{aligned}
\Phi(S) = \frac{\text{flow}(S, \bar{S})}{\pi(S)} &\geq ce^{-\alpha k^2\delta^2} \frac{\alpha(k-1)\delta}{ce^{-\alpha((k-1)\delta)^2}} \\
&\geq \Omega(\alpha(k-1)\delta e^{-\alpha\delta^2(2k-1)}) \geq \Omega(\delta e^{-O(\alpha\delta)}).
\end{aligned}$$

For $\delta < \frac{1}{\alpha}$, we have $\alpha\delta < 1$, so $e^{-O(\alpha\delta)} = \Omega(1)$, thus, $\Phi(S) \geq \Omega(\delta)$. Now, $\pi_{\min} \geq ce^{-\alpha} \geq e^{-1/\delta}$, so $\ln(1/\pi_{\min}) \leq 1/\delta$.

If S is not an interval of the form $[k, 1]$ or $[-1, k]$, then the situation is only better since there is more than one "boundary" point which contributes to flow$(S, \bar{S})$. We do not present this argument here. By Theorem 1.11 in $\Omega(1/\delta^3\varepsilon^2)$ steps, a walk gets within ε of the steady state distribution.

In these examples, we have chosen simple probability distributions. The methods extend to more complex situations.

1.9 Bibliographic Notes

The material on the analogy between random walks on undirected graphs and electrical networks is from [**?**] as is the material on random walks in Euclidean space. Additional material on Markov Chains can be found in [**?**], [**?**], and [**?**]. For material on Markov Chain Monte Carlo methods see [**?**] and [**?**].

The use of Minimum Escape Probability (also called conductance) to prove convergence of Markov Chains is by Sinclair and Jerrum, [**?**] and Alon [**?**]. A polynomial time bounded Markov Chain based method for estimating the volume of convex sets was developed by Dyer, Frieze and Kannan [**?**].

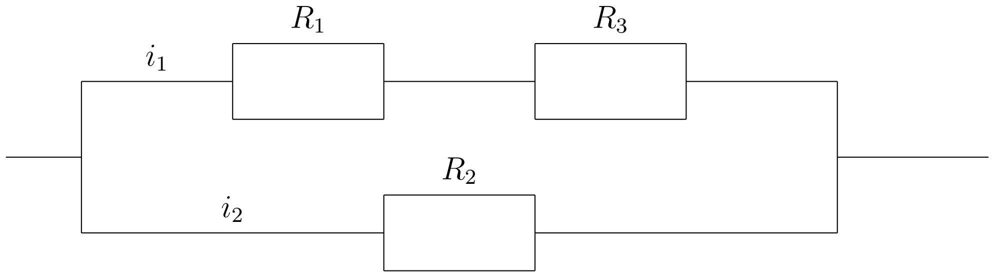


Figure 1.10: An electrical network of resistors.

1.10 Exercises

Exercise 1.1

a *Give an example of a graph, with cycles of more than one length, for which the greatest common divisor of all cycle lengths is three.*

b *Prove that a graph is bipartite if and only if it has no odd length cycle.*

c *Show that for the random walk on a bipartite graph (with any edge weights), the steady state probabilities do not exist.*

Exercise 1.2

(a) *What is the set of possible harmonic functions on a graph if there are only interior vertices and no boundary vertices that supply the boundary condition?*

(b) *Let q_x be the steady state probability of vertex x in a random walk on an undirected graph and let d_x be the degree of vertex x. Show that $\frac{q_x}{d_x}$ is a harmonic function.*

(c) *If there are multiple harmonic functions when there are no boundary conditions why is the steady state probability of a random walk on an undirected graph unique?*

(d) *What is the steady state probability of a random walk on an undirected graph?*

Exercise 1.3 *Consider the electrical resistive network in Figure 1.10 consisting of vertices connected by resistors. Kirchoff's law states that the currents at each node sum to zero. Ohm's law states that the voltage across a resistor equals the product of the resistance times the current through it. Using these laws calculate the effective resistance of the network.*

Solution:

$\frac{(r_1+r_3)r_2}{r_1+r_2+r_3}$

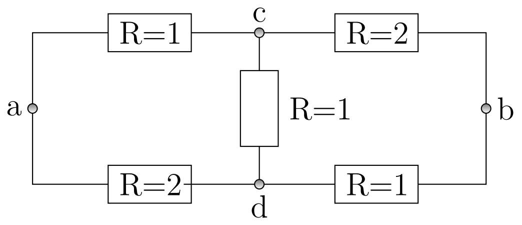


Figure 1.11: An electrical network of resistors.

Exercise 1.4 *Given a graph consisting of a single path of five vertices numbered 1 to 5, what is the probability of reaching vertex 1 before vertex 5 when starting at vertex 4.*

Exercise 1.5 *Consider the electrical network of Figure 1.11.*

(a) *Set the voltage at a to one and at b to zero. What are the voltages at c and d?*

(b) *What is the current in the edges a to c, a to d, c to d. c to b and d to b?*

(c) *What is the effective resistance between a and b?*

(d) *Convert the electrical network to a graph. What are the edge probabilities at each vertex?*

(e) *What is the probability of a walk starting at c reaching a before b? a walk starting at d?*

(f) *How frequently does a walk from a to b go through the edge from c to d?*

(g) *What is the probability that a random walk starting at a will return to a before reaching b?*

Exercise 1.6 *Prove that the escape probability $p_{escape} = \frac{c_{eff}}{c_a}$ must be less than or equal to one.*

Exercise 1.7 *Prove that reducing the value of a resistor in a network cannot increase the effective resistance. Prove that increasing the value of a resistor cannot decrease the effective resistance.*

Exercise 1.8 *The energy dissipated by the resistance of edge xy in an electrical network is given by $i_{xy}^2 r_{xy}$. The total energy dissipation in the network is $E = \frac{1}{2}\sum\limits_{x,y} i_{xy}^2 r_{xy}$ where the $\frac{1}{2}$ accounts for the fact that the dissipation in each edge is counted twice in the summation. Show that the actual current distribution is that distribution satisfying Ohm's law that minimizes energy dissipation.*

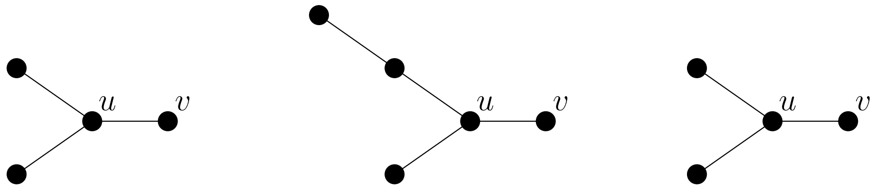


Figure 1.12: Three graphs

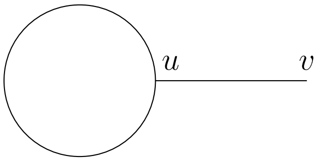


Figure 1.13: A graph consisting of a circle of edges along with a path of length m

Exercise 1.9 *What is the hitting time h_{uv} for two adjacent vertices on a cycle of length n? What is the hitting time if the edge (u, v) is removed?*

Exercise 1.10 *What is the hitting time h_{uv} for the three graphs if Figure 1.14.*

Exercise 1.11 *Consider the n node connected graph shown in Figure 1.13 consisting of an edge (u, v) plus a connected graph on $n-1$ vertices and some number of edges. Prove that $h_{uv} = 2m - 1$ where m is the number of edges in the $n-1$ vertex subgraph.*

Exercise 1.12 *What is the most general solution to the difference equation $t(i+2) - 5t(i+1) + 6t(i) = 0$/ How many boundary conditions do you need to make the solution unique?*

Exercise 1.13 *Given the difference equation $a_k t(i+k) + at(i+k-1) + \cdots + a_1 t(i+1) + a_0 t(i) = 0$ the polynomial $a_k t^k + a_{k-i} t^{k-1} + \cdots + a_1 t + a_0 = 0$ is called the characteristic polynomial.*

(a) *If the equation has a set of r distinct roots, what is the most general form of the solution?*

(b) *If the roots of the characteristic polynomial are not unique what is the most general form of the solution?*

(c) *What is the dimension of the solution spaced?*

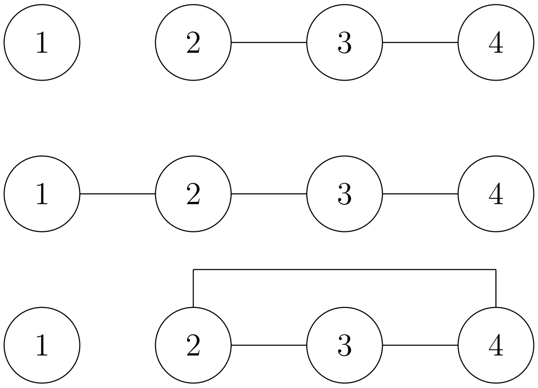


Figure 1.14: Three graph

(d) *If the difference equation is not homogeneous and f(i) is a specific solution to the non homogeneous difference equation, what is the full set of solutions to the difference equation?*

Exercise 1.14 *Consider the set of integers $\{1, 2, \ldots, n\}$. How many draws d with replacement are necessary so that every integer is drawn?*

Exercise 1.15 *Consider a random walk on a clique of size n. What is the expected number of steps before a given vertex is reached?*

Solution:

$$\frac{1}{n} + 2\frac{1}{n}(1 - \frac{1}{n}) + 3\frac{1}{n}(1 - \frac{1}{n}^{2}) + \cdots$$

$$= \frac{1}{n}(1 - \frac{1}{n}) \left[1 + 2(1 - \frac{1}{n}) + 3(1 - \frac{1}{n})^2 + \cdots\right.$$

$$= \frac{1}{n}\left[\frac{\frac{1}{n}}{1-(1-\frac{1}{n})^2}\right] = \frac{1}{n} \quad \frac{1-\frac{1}{n}}{(\frac{1}{n})^2}$$

$$n-1$$

Exercise 1.16 *Show that adding an edge can either increase or decrease hitting time by calculating h_{24} for the three graphs in figure 1.14.*

Exercise 1.17 *Show that adding an edge to a graph can either increase or decrease commute time.*

Exercise 1.18 *Prove that two independent random walks on a two dimensional lattice will hit with probability one.*

Exercise 1.19 *Consider the lattice in 2-dimensions. In each square add the two diagonal edges. What is the escape probability for the resulting graph?*

Exercise 1.20 *Determine by simulation the escape probability for the 3-dimensional lattice.*

Exercise 1.21 *What is the escape probability for a random walk starting at the root of a binary tree?*

Exercise 1.22 *Consider a random walk on the positive half line, that is the integers* $1, 2, 3, \ldots$. *At the origin, always move right one step. At all other integers move right with probability 2/3 and left with probability 1/3. What is the escape probability?*

Exercise 1.23 *What is the probability of returning to the start vertex on a random walk on an infinite planar graph?*

Exercise 1.24 *Create a model for a graph similar to a 3-dimensional lattice in the way that a planar graph is similar to a 2-dimensional lattice. What is probability of returning to the start vertex in your model?*

Exercise 1.25 *Consider a strongly connected directed graph. In the steady state calculate the flow through each edge of a random walk.*

Exercise 1.26 *Create a random directed graph with 200 nodes and roughly eight edges per node. Add* k *new nodes and calculate the page rank with and without directed edges from the* k *added nodes to node 1. How much does adding the* k *edges change the page rank of nodes for various values of* k *and restart frequency? How much does adding a loop at node 1 change the page rank? To do the experiment carefully one needs to consider the page rank of a node to which the star is attached. If it has low page rank its page rank is likely to increase a lot.*

Exercise 1.27 *Repeat the experiment in Exercise 1.26 for hitting time.*

Exercise 1.28 *Search engines ignore self loops in calculating page rank. Thus, to increase page rank one needs to resort to loops of length two. By how much can you increase the page rank of a page by adding a number of loops of length two?*

Exercise 1.29 *Can one increase the page rank of a node* v *in a directed graph by doing something some distance from* v*? The answer is yes if there is a long narrow chain of nodes into* v *with no edges leaving the chain. What if there is no such chain?*

Exercise 1.30 *Consider modifying personal page rank as follows. Start with the uniform restart distribution and calculate the steady state probabilities. Then run the personalized page rank algorithm using the steady state distribution calculated instead of the uniform distribution. Keep repeating until the process converges. That is, we get a steady state probability distribution such that if we use the steady state probability distribution for the restart distribution we will get the steady state probability distribution back. Does this process converge? What is the resulting distribution? What distribution do we get for the graph consisting of two vertices* u *and* v *with a single edge from* u *to* v*?*

Exercise 1.31

(a) What is the hitting time for a vertex in a complete directed graph with self loops?

(b) What is the hitting time for a vertex in a directed cycle with n nodes?

Exercise 1.32 *Using a web browser bring up a web page and look at the source html. How would you extract the url's of all hyperlinks on the page if you were doing a crawl of the web? With Internet Explorer click on "source" under "view" to access the html representation of the web page. With Firefox click on "page source" under "view".*

Exercise 1.33 *Sketch an algorithm to crawl the World Wide Web. There is a time delay between the time you seek a page and the time you get it. Thus, you cannot wait until the page arrives before starting another fetch. There are conventions that must be obeyed if one were to actually do a search. Sites specify information has to how long or which files can be searched. Do not attempt an actual search without guidance from a knowledgeable person.*

Exercise 1.34 *Prove Proposition 1.9 that for two probability distributions $\mathbf{p}, \mathbf{q}$, $|\mathbf{p}-\mathbf{q}|_1 = 2\sum_i (p_i - q_i)^+$.*

Exercise 1.35 *Suppose S is a subset of at most $n^2/2$ points in the $n \times n$ lattice. Show that*

$$|\{(i,j) \in S : \textit{row } i,\ \textit{col. } j \subseteq S\}| \leq |S|/2.$$

Exercise 1.36 *Show that the steady state probabilities of the chain described in the Gibbs sampler is the correct p.*

Exercise 1.37 *A Markov chain is said to be symmetric if for all i and j, $p_{ij} = p_{ji}$. What is the steady state distribution of a connected aperiodic symmetric chain? Prove your answer.*

Exercise 1.38 *How would you integrate a multivariate polynomial distribution over some region?*

Exercise 1.39 *Given a time-reversible Markov chain, modify the chain as follows. At the current state, stay put (no move) with probability 1/2. With the other probability 1/2, move as in the old chain. Show that the new chain has the same steady state. What happens to the convergence time in this modification?*

Exercise 1.40 *Using the Metropolis-Hasting Algorithm create a Markov chain whose stationary probability is that given in the following table.*

x_1x_2	00	01	02	10	11	12	20	21	22
Prob	1/16	1/8	1/16	1/8	1/4	1/8	1/16	1/8	1/16

Exercise 1.41 *Let* $\mathbf{p}$ *be a probability vector (nonnegative components adding up to 1) on the vertices of a connected, aperiodic graph. Set* p_{ij} *(the transition probability from* i *to* j*) to* p_j *for all* $i \neq j$ *which are adjacent in the graph. Show that the steady state probability vector for the chain is* $\mathbf{p}$*. Is running this chain an efficient way to sample according to a distribution close to* $\mathbf{p}$*? Think, for example, of the graph* G *being the* $n \times n \times n \times \cdots n$ *grid.*

Exercise 1.42 *Construct the edge probability for a three state Markov chain so that the steady state probability is* $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$.

Exercise 1.43 *Consider a three state Markov chain with steady state probability* $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$*. Consider the Metropolis-Hastings algorithm with* G *the complete graph on these three nodes. What is the expected probability that we would actually make a move along a selected edge?*

Exercise 1.44 *Try Gibbs sampling on* $p\left(x\right) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$.

What happens? How does the Metropolis Hasting Algorithm do?

Exercise 1.45 *Consider* $p(\mathbf{x})$*, where,* $\mathbf{x} = (x_1, \ldots, x_{100})$ *and* $p\left(\mathbf{0}\right) = \frac{1}{2}$, $p\left(\mathbf{x}\right) = \frac{1}{2^{100}} \quad x \neq 0$*. How does Gibbs sampling behave?*

Exercise 1.46 *Construct an algorithm and compute the volume of a unit radius sphere in 20 dimensions by carrying out a random walk on a* 20×20 *grid with 0.1 spacing.*

Exercise 1.47 *Given a graph* G *and an integer* k *how would you generate a sequence of connected subgraphs* $S_1, S_2, \ldots$ *of* G *of size* k *where* S_i *is generated with probability proportional to* $2^{|E_i|}$ *where* E_i *is the set of edges in* S_i*?*

Exercise 1.48 *What is the mixing time for*

(a) *a clique?*

(b) *two cliques connected by a single edge?*

Exercise 1.49 *Show that for the* $n \times n \times \cdots \times n$ *grid in* d *space, MEP is* $\Omega(1/dn)$*.*
Hint: The argument is a generalization of the argument in Exercise 1.35. Argue that for any subset S *containing at most* $1/2$ *the grid points, for at least* $1/2$ *the grid points in* S*, among the* d *coordinate lines through the point, at least one intersects* $\bar{S}$*.*