4 Random Walks and Markov Chains

A random walk on a directed graph consists of a sequence of vertices generated from a start vertex by probabilistically selecting an incident edge, traversing the edge to a new vertex, and repeating the process.

We generally assume the graph is *strongly connected*, meaning that for any pair of vertices x and y, the graph contains a path of directed edges starting at x and ending at y. If the graph is strongly connected, then, as we will see, no matter where the walk begins the fraction of time the walk spends at the different vertices of the graph converges to a stationary probability distribution.

Start a random walk at a vertex x and think of the starting probability distribution as putting a mass of one on x and zero on every other vertex. More generally, one could start with any probability distribution $\mathbf{p}$, where $\mathbf{p}$ is a row vector with nonnegative components summing to one, with p_x being the probability of starting at vertex x. The probability of being at vertex x at time $t+1$ is the sum over each adjacent vertex y of being at y at time t and taking the transition from y to x. Let $\mathbf{p(t)}$ be a row vector with a component for each vertex specifying the probability mass of the vertex at time t and let $\mathbf{p(t+1)}$ be the row vector of probabilities at time $t+1$. In matrix notation[14]

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

where the ij^{th} entry of the matrix P is the probability of the walk at vertex i selecting the edge to vertex j.

A fundamental property of a random walk is that in the limit, the long-term average probability of being at a particular vertex is independent of the start vertex, or an initial probability distribution over vertices, provided only that the underlying graph is strongly connected. The limiting probabilities are called the *stationary probabilities*. This fundamental theorem is proved in the next section.

A special case of random walks, namely random walks on undirected graphs, has important connections to electrical networks. Here, each edge has a parameter called *conductance*, like electrical conductance. If the walk is at vertex x, it chooses an edge to traverse next from among all edges incident to x with probability proportional to its conductance. Certain basic quantities associated with random walks are hitting time, which is the expected time to reach vertex y starting at vertex x, and cover time, which is the expected time to visit every vertex. Qualitatively, for undirected graphs these quantities are all bounded above by polynomials in the number of vertices. The proofs of these facts will rely on the analogy between random walks and electrical networks.

[14] Probability vectors are represented by row vectors to simplify notation in equations like the one here.

random walk	**Markov chain**
graph	stochastic process
vertex	state
strongly connected	persistent
aperiodic	aperiodic
strongly connected and aperiodic	ergodic
undirected graph	time reversible

Table 5.1: Correspondence between terminology of random walks and Markov chains

Aspects of the theory of random walks were developed in computer science with a number of applications. Among others, these include defining the pagerank of pages on the World Wide Web by their stationary probability. An equivalent concept called a *Markov chain* had previously been developed in the statistical literature. A Markov chain has a finite set of *states*. For each pair of states x and y, there is a *transition 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 starts at some state. At a given time step, if it is in state x, the next state y is selected randomly with probability p_{xy}. A Markov chain can be represented by a directed graph with a vertex representing each state and an edge with weight p_{xy} from vertex x to vertex y. We say that the Markov chain is *connected* if the underlying directed graph is strongly connected. That is, if there is a directed path from every vertex to every other vertex. The matrix P consisting of the p_{xy} is called the *transition probability matrix* of the chain. The terms "random walk" and "Markov chain" are used interchangeably. The correspondence between the terminologies of random walks and Markov chains is given in Table 5.1.

A state of a Markov chain 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 is equivalent to the property that the state is in a strongly connected component with no out edges. For most of the chapter, we assume that the underlying directed graph is strongly connected. We discuss here briefly what might happen if we do not have strong connectivity. Consider the directed graph in Figure 4.1b with three strongly connected components, A, B, and C. Starting from any vertex in A, there is a nonzero probability of eventually reaching any vertex in A. However, the probability of returning to a vertex in A is less than one and thus vertices in A, and similarly vertices in B, are not persistent. From any vertex in C, the walk eventually will return with probability one to the vertex, since there is no way of leaving component C. Thus, vertices in C are persistent.

A connected Markov Chain is said to be *aperiodic* if the greatest common divisor of the lengths of directed cycles is one. It is known that for connected aperiodic chains, the

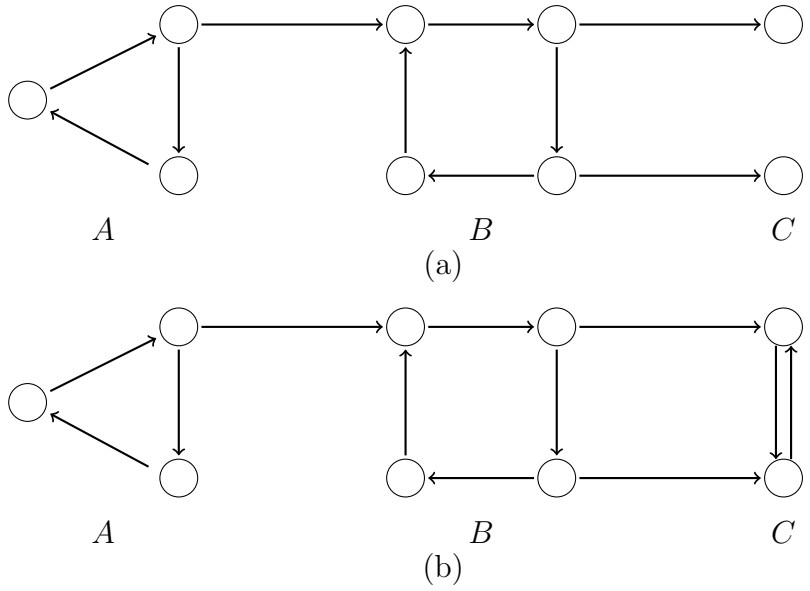


Figure 4.1: (a) A directed graph with vertices having no out out edges and a strongly connected component A with no in edges.
(b) A directed graph with three strongly connected components.

probability distribution of the random walk converges to a unique stationary distribution. Aperiodicity is a technical condition needed in this proof. Here, we do not prove this theorem and do not worry about aperiodicity at all. It turns out that if we take the average probability distribution of the random walk over the first t steps, then this average converges to a limiting distribution for connected chains (without assuming aperiodicity) and this average is what one uses in practice. We prove this limit theorem and explain its uses in what is called the Markov Chain Monte Carlo (MCMC) method.

Markov chains are used to model situations where all the information of the system necessary to predict the future can be encoded in the current state. A typical example is speech, where for a small k the current state encodes the last k syllables uttered by the speaker. Given the current state, there is a certain probability of each syllable being uttered next and these can be used to calculate the transition probabilities. Another example is a gambler's assets, which 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 gambler's bets depend only on current assets, not the past history.

Later in the chapter, we study the widely used Markov Chain Monte Carlo method (MCMC). Here, the objective is to sample a large space according to some probability distribution p. The number of elements in the space may be very large, say 10^{100}. One designs a Markov chain where states correspond to the elements of the space. The transition probabilities of the chain are designed so that the stationary probability of the chain is the

probability distribution p with which we want to sample. One chooses samples by taking a random walk until the probability distribution is close to the stationary distribution of the chain and then selects the current state of the walk. The walk continues a number of steps until the probability distribution is nearly independent of where the walk was when the first element was selected. A second point is then selected, and so on. Although it is impossible to store the graph in a computer since it has 10^{100} vertices, to do the walk one needs only store the current vertex of the walk and be able to generate the adjacent vertices by some algorithm. What is critical is that the probability distribution of the walk converges to the stationary distribution in time logarithmic in the number of states.

We mention two motivating examples. The first is to select a point at random in d-space according to a probability density such as a Gaussian. Put down a grid and let each grid point be a state of the Markov chain. Given a probability density p, design transition probabilities of a Markov chain so that the stationary distribution is p. In general, the number of states grows exponentially in the dimension d, but if the time to converge to the stationary distribution grows polynomially in d, then one can do a random walk on the graph until convergence to the stationary probability. Once the stationary probability has been reached, one selects a point. To select a set of points, one must walk a number of steps between each selection so that the probability of the current point is independent of the previous point. By selecting a number of points one can estimate the probability of a region by observing the number of selected points in the region.

A second example is from physics. Consider an $n \times n$ grid in the plane with a particle at each grid point. Each particle has a spin of ± 1. A configuration is a n^2 dimensional vector $\mathbf{v} = (v_1, v_2, \ldots, v_{n^2})$, where v_i is the spin of the i^{th} particle. There are 2^{n^2} spin configurations. The energy of a configuration is a function $f(\mathbf{v})$ of the configuration, not of any single spin. A central problem in statistical mechanics is to sample spin configurations according to their probability. It is easy to design a Markov chain with one state per spin configuration so that the stationary probability of a state is proportional to the state's energy. If a random walk gets close to the stationary probability in time polynomial in n rather than 2^{n^2}, then one can sample spin configurations according to their probability.

The Markov Chain has 2^{n^2} states, one per configuration. Two states in the Markov chain are adjacent if and only if the corresponding configurations $\mathbf{v}$ and $\mathbf{u}$ differ in just one coordinate ($u_i = v_i$ for all but one i). The Metropilis-Hastings random walk, described in more detail in Section 4.2, has a transition probability from a configuration $\mathbf{v}$ to an adjacent configuration $\mathbf{u}$ of

$$\frac{1}{n^2} \min \left(1, \frac{f(\mathbf{u})}{f(\mathbf{v})}\right).$$

As we will see, the Markov Chain has a stationary probability proportional to the energy. There are two more crucial facts about this chain. The first is that to execute a step in the chain, we do not need the whole chain, just the ratio $\frac{f(\mathbf{u})}{f(\mathbf{v})}$. The second is that under suitable assumptions, the chain approaches stationarity in time polynomial in n.

A quantity called the *mixing time*, loosely defined as the time needed to get close to the stationary distribution, is often much smaller than the number of states. In Section 4.4, we relate the mixing time to a combinatorial notion called *normalized conductance* and derive upper bounds on the mixing time in several cases.

4.1 Stationary Distribution

Let $\mathbf{p}(\mathbf{t})$ be the probability distribution after t steps of a random walk. Define the *long-term average probability distribution* $\mathbf{a}(\mathbf{t})$ by

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

The fundamental theorem of Markov chains asserts that for a connected Markov chain, $\mathbf{a}(\mathbf{t})$ converges to a limit probability vector $\mathbf{x}$ that satisfies the equations $\mathbf{x}P = \mathbf{x}$. Before proving the fundamental theorem of Markov chains, we first prove a technical lemma.

Lemma 4.1 *Let P be the transition probability matrix for a connected Markov chain. The $n \times (n+1)$ matrix $A = [P - I\ ,\ \mathbf{1}]$ obtained by augmenting the matrix $P - I$ with an additional column of ones has rank n.*

Proof: If the rank of $A = [P - I, \mathbf{1}]$ was less than n there would be a subspace of solutions to $A\mathbf{x} = \mathbf{0}$ of at least two-dimensions. Each row in P sums to one, so each row in $P - I$ sums to zero. Thus $\mathbf{x} = (\mathbf{1}, 0)$, where all but the last coordinate of $\mathbf{x}$ is 1, is one solution to $A\mathbf{x} = \mathbf{0}$. Assume there was a second solution $(\mathbf{x}, \alpha)$ perpendicular to $(\mathbf{1}, 0)$. Then $(P-I)\mathbf{x}+\alpha\mathbf{1} = \mathbf{0}$ and for each i, $x_i = \sum_j p_{ij}x_j + \alpha$. Each x_i is a convex combination of some x_j plus α. Let S be the set of i for which x_i attains its maximum value. Since $\mathbf{x}$ is perpendicular to $\mathbf{1}$, some x_i is negative and thus $\bar{S}$ is not empty. Connectedness implies that some x_k of maximum value is adjacent to some x_l of lower value. Thus, $x_k > \sum_j p_{kj}x_j$. Therefore α must be greater than 0 in $x_k = \sum_j p_{kj}x_j + \alpha$..

On the other hand, the same argument with T the set of i with x_i taking its minimum value implies $\alpha < 0$. This contradiction falsifies the assumption of a second solution, thereby proving the lemma. ∎

Theorem 4.2 (Fundamental Theorem of Markov Chains) *For a connected Markov chain there is a unique probability vector $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi}P = \boldsymbol{\pi}$. Moreover, for any starting distribution, $\lim_{t\to\infty} \mathbf{a}(\mathbf{t})$ exists and equals $\boldsymbol{\pi}$.*

Proof: Note that $\mathbf{a}(\mathbf{t})$ is itself a probability vector, since its components are nonnegative and sum to 1. Run one step of the Markov chain starting with distribution $\mathbf{a}(\mathbf{t})$; the

distribution after the step is $\mathbf{a}(\mathbf{t})P$. Calculate the change in probabilities due to this step.

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

Thus, $\mathbf{b}(\mathbf{t}) = \mathbf{a}(\mathbf{t})P - \mathbf{a}(\mathbf{t})$ satisfies $|\mathbf{b}(\mathbf{t})| \leq \frac{2}{t} \to 0$, as $t \to \infty$.

By Lemma 4.1 above, $A = [P - I, \mathbf{1}]$ has rank n. The $n \times n$ submatrix B of A consisting of all its columns except the first is invertible. Let $\mathbf{c}(\mathbf{t})$ be obtained from $\mathbf{b}(\mathbf{t})$ by removing the first entry. Since $\mathbf{a}(\mathbf{t})P - \mathbf{a}(\mathbf{t}) = \mathbf{b}(\mathbf{t})$ and B is obtained by deleting the first column of $P - I$ and adding a column of 1's, $\mathbf{a}(\mathbf{t})B = [\mathbf{c}(\mathbf{t}), 1]$. Then $\mathbf{a}(\mathbf{t}) = [\mathbf{c}(\mathbf{t})\ ,\ 1]B^{-1} \to [\mathbf{0}\ ,\ 1]B^{-1}$ establishing the theorem with $\boldsymbol{\pi} = [\mathbf{0}\ ,\ 1]B^{-1}$. ∎

We finish this section with the following lemma useful in establishing that a probability distribution is the stationary probability distribution for a random walk on a connected graph with edge probabilities.

Lemma 4.3 *For a random walk on a strongly connected graph with probabilities on the edges, if the vector $\boldsymbol{\pi}$ satisfies $\pi_x p_{xy} = \pi_y p_{yx}$ for all x and y and $\sum_x \pi_x = 1$, then $\boldsymbol{\pi}$ is the stationary distribution of the walk.*

Proof: Since $\boldsymbol{\pi}$ satisfies $\pi_x p_{xy} = \pi_y p_{yx}$, summing both sides, $\pi_x = \sum_y \pi_y p_{yx}$ and hence $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi} = \boldsymbol{\pi}P$. By Theorem 4.2, $\boldsymbol{\pi}$ is the unique stationary probability. ∎

4.2 Markov Chain Monte Carlo

The Markov Chain Monte Carlo (MCMC) method is a technique for sampling a multivariate probability distribution $p(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$. The MCMC method is used to estimate the expected value of a function $f(\mathbf{x})$

$$E(f) = \sum_{\mathbf{x}} f(\mathbf{x})p(\mathbf{x}).$$

If each x_i can take on two or more values, then there are at least 2^d values for $\mathbf{x}$, so an explicit summation requires exponential time. Instead, one could draw a set of samples, where each sample $\mathbf{x}$ is selected with probability $p(\mathbf{x})$. Averaging f over these samples provides an estimate of the sum.

To sample according to $p(\mathbf{x})$, design a Markov Chain whose states correspond to the possible values of $\mathbf{x}$ and whose stationary probability distribution is $p(\mathbf{x})$. There are two general techniques to design such a Markov Chain: the Metropolis-Hastings algorithm

and Gibbs sampling, which we will describe in the next two subsections. The Fundamental Theorem of Markov Chains, Theorem 4.2, states that the average of the function f over states seen in a sufficiently long run is a good estimate of $E(f)$. The harder task is to show that the number of steps needed before the long-run average probabilities are close to the stationary distribution grows polynomially in d, though the total number of states may grow exponentially in d. This phenomenon known as *rapid mixing* happens for a number of interesting examples. Section 4.4 presents a crucial tool used to show rapid mixing.

We used $\mathbf{x} \in \mathbf{R}^d$ to emphasize that distributions are multi-variate. From a Markov chain perspective, each value $\mathbf{x}$ can take on is a state, i.e., a vertex 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, vertices of the graph are called states.

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

$$\mathbf{a}(\mathbf{t}) = \frac{1}{t}\left[\mathbf{p}(\mathbf{0}) + \mathbf{p}(\mathbf{1}) + \cdots + \mathbf{p}(\mathbf{t-1})\right]. \tag{4.1}$$

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

$$E(\gamma) = \sum_i f_i \left(\frac{1}{t} \sum_{j=1}^{t} \text{Prob}\big(\text{walk is in state } i \text{ at time } j\big) \right) = \sum_i f_i a_i(t).$$

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

$$\left| \sum_i f_i p_i - E(\gamma) \right| \leq f_{\max} \sum_i |p_i - a_i(t)| = f_{\max} ||\mathbf{p} - \mathbf{a}(\mathbf{t})||_1 \tag{4.2}$$

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

The following proposition is often useful.

Proposition 4.4 *For two probability distributions* $\mathbf{p}$ *and* $\mathbf{q}$*,*

$$||\mathbf{p}-\mathbf{q}||_1 = 2\sum_i (p_i - q_i)^+ = 2\sum_i (q_i - p_i)^+$$

where $x^+ = x$ *if* $x \geq 0$ *and* $x^+ = 0$ *if* $x < 0$*.*

The proof is left as an exercise.

4.2.1 Metropolis-Hasting Algorithm

The Metropolis-Hasting algorithm is a general method to design a Markov chain whose stationary distribution is a given target distribution $\mathbf{p}$. Start with a connected undirected graph G on the set of states. 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 could be the lattice graph with $2d$ coordinate edges at each interior vertex. In general, let r be the maximum degree of any vertex 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_i values. For i adjacent to j in G,

$$p_{ij} = \frac{1}{r}\min\left(1, \frac{p_j}{p_i}\right)$$

and

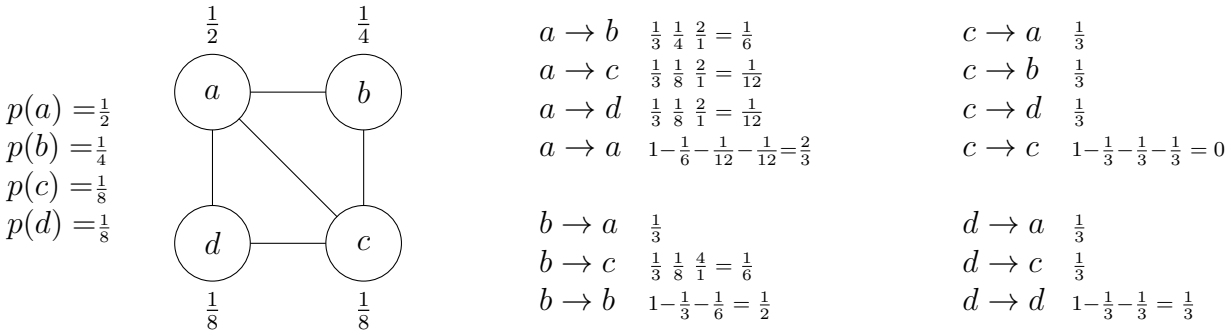
$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}.$$

Thus,

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

By Lemma 4.3, the stationary probabilities are indeed p_i as desired.

Example: Consider the graph in Figure 4.2. 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 the desired probability. ∎



$$p(a) = p(a)p(a \to a) + p(b)p(b \to a) + p(c)p(c \to a) + p(d)p(d \to a)$$
$$= \tfrac{1}{2}\ \tfrac{2}{3} + \tfrac{1}{4}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{2}$$

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

$$p(c) = p(a)p(a \to c) + p(b)p(b \to c) + p(c)p(c \to c) + p(d)p(d \to c)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{4}\ \tfrac{1}{6} + \tfrac{1}{8}\ 0 + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

$$p(d) = p(a)p(a \to d) + p(c)p(c \to d) + p(d)p(d \to d)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

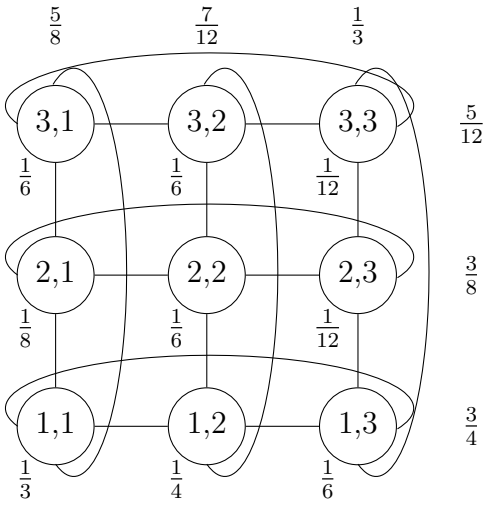
Figure 4.2: Using the Metropolis-Hasting algorithm to set probabilities for a random walk so that the stationary probability will be the desired probability.

4.2.2 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 an undirectd 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. Thus, the underlying graph is like a d-dimensional lattice except that the vertices in the same coordinate line form a clique.

To generate samples of $\mathbf{x} = (x_1, \ldots, x_d)$ with a target distribution $p(\mathbf{x})$, 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. Without loss



$p(1,1) = \frac{1}{3}$
$p(1,2) = \frac{1}{4}$
$p(1,3) = \frac{1}{6}$
$p(2,1) = \frac{1}{8}$
$p(2,2) = \frac{1}{6}$
$p(2,3) = \frac{1}{12}$
$p(3,1) = \frac{1}{6}$
$p(3,2) = \frac{1}{6}$
$p(3,3) = \frac{1}{12}$

$p_{(11)(12)} = \frac{1}{d} p_{12}/(p_{11} + p_{12} + p_{13}) = \frac{1}{2}\ (\frac{1}{4})/(\frac{1}{3}\ \frac{1}{4}\ \frac{1}{6}) = \frac{1}{8}/\frac{9}{12} = \frac{1}{8}\ \frac{4}{3} = \frac{1}{6}$
Calculation of edge probability $p_{(11)(12)}$

$p_{(11)(12)} = \frac{1}{2}\ \frac{1}{4}\ \frac{4}{3} = \frac{1}{6}$ $p_{(12)(11)} = \frac{1}{2}\ \frac{1}{3}\ \frac{4}{3} = \frac{2}{9}$ $p_{(13)(11)} = \frac{1}{2}\ \frac{1}{3}\ \frac{4}{3} = \frac{2}{9}$ $p_{(21)(22)} = \frac{1}{2}\frac{1}{6}\frac{8}{3} = \frac{2}{9}$
$p_{(11)(13)} = \frac{1}{2}\ \frac{1}{6}\ \frac{4}{3} = \frac{1}{9}$ $p_{(12)(13)} = \frac{1}{2}\ \frac{1}{6}\ \frac{4}{3} = \frac{1}{9}$ $p_{(13)(12)} = \frac{1}{2}\ \frac{1}{4}\ \frac{4}{3} = \frac{1}{6}$ $p_{(21)(23)} = \frac{1}{2}\ \frac{1}{12}\ \frac{8}{3} = \frac{1}{9}$
$p_{(11)(21)} = \frac{1}{2}\ \frac{1}{8}\ \frac{8}{5} = \frac{1}{10}$ $p_{(12)(22)} = \frac{1}{2}\ \frac{1}{6}\ \frac{12}{7} = \frac{1}{7}$ $p_{(13)(23)} = \frac{1}{2}\ \frac{1}{12}\ \frac{3}{1} = \frac{1}{8}$ $p_{(21)(11)} = \frac{1}{2}\frac{1}{3}\frac{8}{5} = \frac{4}{15}$
$p_{(11)(31)} = \frac{1}{2}\ \frac{1}{6}\ \frac{8}{5} = \frac{2}{15}$ $p_{(12)(32)} = \frac{1}{2}\ \frac{1}{6}\ \frac{12}{7} = \frac{1}{7}$ $p_{(13)(33)} = \frac{1}{2}\ \frac{1}{12}\ \frac{3}{1} = \frac{1}{8}$ $p_{(21)(31)} = \frac{1}{2}\frac{1}{6}\frac{8}{5} = \frac{2}{15}$
Edge probabilities.

$p_{11}p_{(11)(12)} = \frac{1}{3}\ \frac{1}{6} = \frac{1}{4}\ \frac{2}{9} = p_{12}p_{(12)(11)}$
$p_{11}p_{(11)(13)} = \frac{1}{3}\ \frac{1}{9} = \frac{1}{6}\ \frac{2}{9} = p_{13}p_{(13)(11)}$
$p_{11}p_{(11)(21)} = \frac{1}{3}\ \frac{1}{10} = \frac{1}{8}\ \frac{4}{15} = p_{21}p_{(21)(11)}$
Verification of a few edges, $p_i p_{ij} = p_j p_{ji}$.

Note that the edge probabilities out of a state such as (1,1) do not add up to one. That is, with some probability the walk stays at the state that it is in. For example, $p_{(11)(11)} = 1 - (p_{(11)(12)} + p_{(11)(13)} + p_{(11)(21)} + p_{(11)(31)}) = 1 - \frac{1}{6} - \frac{1}{24} - \frac{1}{32} - \frac{1}{24} = \frac{9}{32}$.

Figure 4.3: Using the Gibbs algorithm to set probabilities for a random walk so that the stationary probability will be a desired probability.

of generality let that coordinate be the first. 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

$$p_{\mathbf{xy}} = \frac{1}{d} p(y_1 | x_2, x_3, \ldots, x_d).$$

The normalizing constant is $1/d$ since $\sum_{y_1} p(y_1 | x_2, x_3, \ldots, x_d)$ equals 1 and summing over d coordinates

$$\sum_{i=1}^{d} \sum_{y_i} p(y_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1} \ldots x_d) = d$$

gives a value of d. Similarly,

$$\begin{aligned} p_{\mathbf{yx}} =& \frac{1}{d} p(x_1 | y_2, y_3, \ldots, y_d) \\ =& \frac{1}{d} p(x_1 | x_2, x_3, \ldots, x_d). \end{aligned}$$

Here use was made of the fact that for $j \neq 1$, $x_j = y_j$.

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

$$\begin{aligned} p_{\mathbf{xy}} &= \frac{1}{d} \frac{p(y_1 | x_2, x_3, \ldots, x_d) p(x_2, x_3, \ldots, x_d)}{p(x_2, x_3, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(y_1, x_2, x_3, \ldots, x_d)}{p(x_2, x_3, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(\mathbf{y})}{p(x_2, x_3, \ldots, x_d)} \end{aligned}$$

again using $x_j = y_j$ for $j \neq 1$. Similarly write

$$p_{\mathbf{yx}} = \frac{1}{d} \frac{p(\mathbf{x})}{p(x_2, x_3, \ldots, x_d)}$$

from which it follows that $p(\mathbf{x}) p_{xy} = p(\mathbf{y}) p_{yx}$. By Lemma 4.3 the stationary probability of the random walk is $p(\mathbf{x})$.

4.3 Areas and Volumes

Computing areas and volumes is a classical problem. For many regular figures in two and three dimensions there are closed form formulae. In Chapter 2, we saw how to compute volume of a high dimensional sphere by integration. For general convex sets in d-space, there are no closed form formulae. Can we estimate volumes of d-dimensional convex sets in time that grows as a polynomial function of d? The MCMC method answes

this question in the affirmative.

One way to estimate the area of the region is to enclose it in a rectangle and estimate the ratio of the area of the region to the area of the rectangle by picking random points in the rectangle and seeing what proportion land in the region. Such methods fail in high dimensions. Even for a sphere in high dimension, a cube enclosing the sphere has exponentially larger area, so exponentially many samples are required to estimate the volume of the sphere.

It turns out, however, that the problem of estimating volumes of sets can be reduced to the problem of drawing uniform random samples from sets. Suppose one wants to estimate the volume of a convex set R. Create a concentric series of larger and larger spheres[15] $S_1, S_2, \ldots, S_k$ such that S_1 is contained in R and S_k contains R. Then

$$\text{Vol}(R) = \text{Vol}(S_k \cap R) = \frac{\text{Vol}(S_k \cap R)}{\text{Vol}(S_{k-1} \cap R)} \frac{\text{Vol}(S_{k-1} \cap R)}{\text{Vol}(S_{k-2} \cap R)} \cdots \frac{\text{Vol}(S_2 \cap R)}{\text{Vol}(S_1 \cap R)} \text{Vol}(S_1)$$

If the radius of the sphere S_i is $1 + \frac{1}{d}$ times the radius of the sphere S_{i-1}, then we have:

$$1 \;\leq\; \frac{\text{Vol}(S_i \cap R)}{\text{Vol}(S_{i-1} \cap R)} \;\leq\; e$$

because $\text{Vol}(S_i)/\text{Vol}(S_{i-1}) = \left(1 + \frac{1}{d}\right)^d < e$, and the fraction of S_i occupied by R is less than or equal to the fraction of S_{i-1} occupied by R (due to the convexity of R and the fact that the center of the spheres lies in R). This implies that the ratio $\frac{Vol(S_i \cap R)}{Vol(S_{i-1} \cap R)}$ can be estimated by rejection sampling, i.e., selecting points in $S_i \cap R$ uniformly at random and computing the fraction in $S_{i-1} \cap R$, provided one can select points at random from a d-dimensional convex region.

The number of spheres is at most

$$O(\log_{1+(1/d)} r) = O(rd)$$

where r is the ratio of the radius of S_k to the radius of S_1. This means that it suffices to estimate each ratio to a factor of $(1 \pm \frac{\epsilon}{erd})$ in order to estimate the overall volume to error $1 \pm \epsilon$.

It remains to show how to draw a uniform random sample from a d-dimensional convex set. Here we will use the convexity of the set R and thus the sets $S_i \cap R$ so that the Markov chain technique will converge quickly to its stationary probability. To select a random sample from a d-dimensional convex set, impose 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 $1/(2d)$ and go to the neighbor if it is still in the

[15] One could also use rectangles instead of spheres.

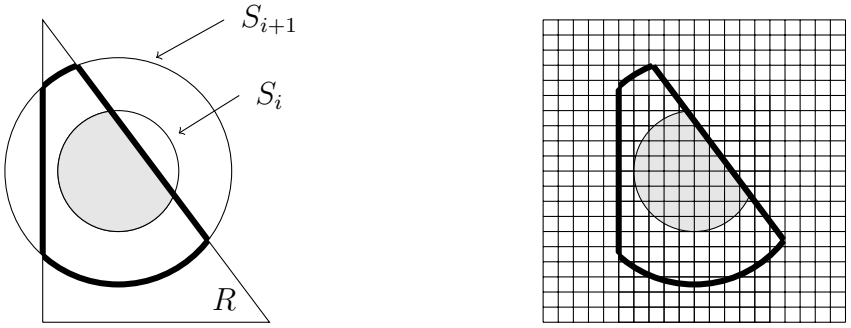


Figure 4.4: By sampling the area inside the dark line and determining the fraction of points in the shaded region we compute $\frac{Vol(S_{i+1} \cap R)}{Vol(S_i \cap R)}$.
To sample we create a grid and assign a probability of one to each grid point inside the dark lines and zero outside. Using Metropolis-Hasting edge probabilities the stationary probability will be uniform for each point inside the region and we can sample points uniformly and determine the fraction within the shaded region.

set; otherwise, stay put and repeat. If the grid length in each of the d coordinate directions is at most some a, the total number of grid points in the set is at most a^d. Although this is exponential in d, the Markov chain turns out to be rapidly mixing (the proof is beyond our scope here) and leads to polynomial time bounded algorithm to estimate the volume of any convex set in $\mathbf{R}^d$.

4.4 Convergence of Random Walks on Undirected Graphs

The Metropolis-Hasting algorithm and Gibbs sampling both involve random walks on edge-weighted undirected graphs. Given an edge-weighted undirected graph, let w_{xy} denote the weight of the edge between nodes x and y, with $w_{xy} = 0$ if no such edge exists. Let $w_x = \sum_y w_{xy}$. The Markov chain has transition probabilities $p_{xy} = w_{xy}/w_x$. We assume the chain is connected.

We now claim that the stationary distribution $\boldsymbol{\pi}$ of this walk has π_x proportional to w_x, i.e., $\pi_x = w_x/w_{total}$ for $w_{total} = \sum_{x'} w_{x'}$. Specifically, notice that

$$w_x p_{xy} = w_x \frac{w_{xy}}{w_x} = w_{xy} = w_{yx} = w_y \frac{w_{yx}}{w_y} = w_y p_{yx}.$$

Therefore $(w_x/w_{total})p_{xy} = (w_y/w_{total})p_{yx}$ and Lemma 4.3 implies that the values $\pi_x = w_x/w_{total}$ are the stationary probabilities.

An important question is how fast the walk starts to reflect the stationary probability of the Markov process. If the convergence time was proportional to the number of states, algorithms such as Metropolis-Hasting and Gibbs sampling would not be very useful since

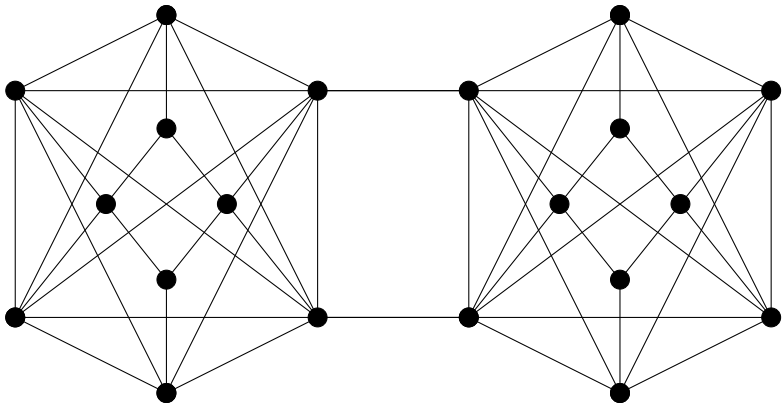


Figure 4.5: A network with a constriction. All edges have weight 1.

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 4.5, takes a long time to converge since the walk is unlikely to cross the narrow passage between the two halves, both of which are reasonably big. We will show in Theorem 4.5 that the time to converge is quantitatively related to the tightest constriction.

We define below a combinatorial measure of constriction for a Markov chain, called the *normalized conductance*. We will relate normalized conductance to the time by which the average probability distribution of the chain is guaranteed to be close to the stationary probability distribution. We call this ε-mixing time:

Definition 4.1 *Fix $\varepsilon > 0$. The* ε-mixing time *of a Markov chain is the minimum integer t such that for any starting distribution $\mathbf{p}$, the 1-norm difference between the t-step running average probability distribution*[16] *and the stationary distribution is at most ε.* ∎

Definition 4.2 *For a subset S of vertices, let $\pi(S)$ denote $\sum_{x\in S}\pi_x$. The normalized conductance $\Phi(S)$ of S is*

$$\Phi(S) = \frac{\sum\limits_{(x,y)\in(S,\bar{S})} \pi_x p_{xy}}{\min\left(\pi(S), \pi(\bar{S})\right)}.$$

∎

There is a simple interpretation of $\Phi(S)$. Suppose without loss of generality that $\pi(S) \leq \pi(\bar{S})$. Then, we may write $\Phi(S)$ as

$$\Phi(S) = \sum_{x\in S} \underbrace{\frac{\pi_x}{\pi(S)}}_{a} \underbrace{\sum_{y\in\bar{S}} p_{xy}}_{b}.$$

[16]Recall that $\mathbf{a}(\mathbf{t}) = \frac{1}{t}\left(\mathbf{p}(\mathbf{0}) + \mathbf{p}(\mathbf{1}) + \cdots + \mathbf{p}(\mathbf{t-1})\right)$ is called the running average distribution.

Here, a is the probability of being in x if we were in the stationary distribution restricted to S and b is the probability of stepping from x to $\bar{S}$ in a single step. Thus, $\Phi(S)$ is the probability of moving from S to $\bar{S}$ in one step if we are in the stationary distribution restricted to S.

It is easy to show that if we started in the distribution $p_{0,x} = \pi_s/\pi(S)$ for $x \in S$ and $p_{0,x} = 0$ for $x \in \bar{S}$, the expected number of steps before we step into $\bar{S}$ is

$$1\Phi(S) + 2(1 - \Phi(S))\Phi(S) + 3(1 - \Phi(S))^2\Phi(S) + \cdots = \frac{1}{\Phi(S)}.$$

Clearly, to be close to the stationary distribution, we must at least get to $\bar{S}$ once. So, mixing time is lower bounded by $1/\Phi(S)$. Since we could have taken any S, mixing time is lower bounded by the minimum over all S of $\Phi(S)$. We define this quantity to be the normalized conductance of the Markov Chain.

Definition 4.3 *The* normalized conductance *of the Markov chain, denoted* Φ, *is defined by*

$$\Phi = \min_{S \subset V, S \neq \{\}} \Phi(S).$$

As we just argued, normalized conductance being high is a necessary condition for rapid mixing. The theorem below proves the converse that normalized conductance being high is sufficient for mixing. Intuitively, if Φ is large, the walk rapidly leaves any subset of states. But the proof of the theorem is quite difficult. After we prove it, 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. In fact for some graphs, called expanders, the mixing time is logarithmic in the number of states.

Theorem 4.5 *The ε-mixing time of a random walk on an undirected graph is*

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

where π_{min} is the minimum stationary probability of any state.

Proof: Let $t = \frac{c\ln(1/\pi_{\min})}{\Phi^2\varepsilon^3}$, for a suitable constant c. Let

$$\mathbf{a} = \mathbf{a(t)} = \frac{1}{t}\big(\mathbf{p(0)} + \mathbf{p(1)} + \cdots + \mathbf{p(t-1)}\big)$$

be the running average distribution. We need to show that $||\mathbf{a} - \boldsymbol{\pi}||_1 \leq \varepsilon$. Let

$$v_i = \frac{a_i}{\pi_i},$$

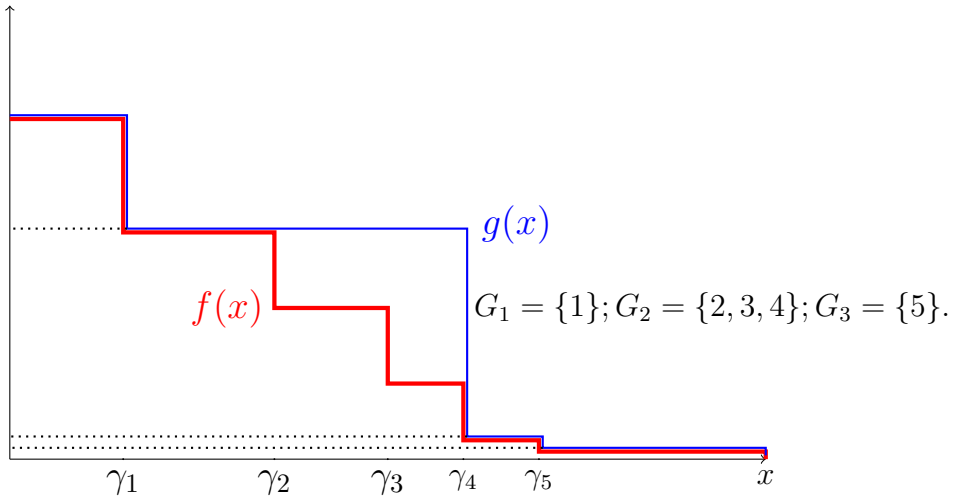


Figure 4.6: Bounding l_1 distance.

and renumber states so that $v_1 \geq v_2 \geq v_3 \geq \cdots$. Thus, early indices i for which $v_i > 1$ are states that currently have too much probability, and late indices i for which $v_i < 1$ are states that currently have too little probability.

Intuitively, to show that $||\mathbf{a} - \boldsymbol{\pi}||_1 \leq \varepsilon$ it is enough to show that the values v_i are relatively flat and do not drop too fast as we increase i. We begin by reducing our goal to a formal statement of that form. Then, in the second part of the proof, we prove that v_i do not fall fast using the concept of "probability flows".

We call a state i for which $v_i > 1$ "heavy" since it has more probability according to $\mathbf{a}$ than its stationary probability. Let i_0 be the maximum i such that $v_i > 1$; it is the last heavy state. By Proposition (4.4):

$$||\mathbf{a} - \boldsymbol{\pi}||_1 \;=\; 2\sum_{i=1}^{i_0}(v_i - 1)\pi_i \;=\; 2\sum_{i \geq i_0+1}(1 - v_i)\pi_i. \tag{4.3}$$

Let

$$\gamma_i = \pi_1 + \pi_2 + \cdots + \pi_i.$$

Define a function $f : [0, \gamma_{i_0}] \to \Re$ by $f(x) = v_i - 1$ for $x \in [\gamma_{i-1}, \gamma_i)$. See Figure 4.6. Now,

$$\sum_{i=1}^{i_0}(v_i - 1)\pi_i = \int_0^{\gamma_{i_0}} f(x)\, dx. \tag{4.4}$$

We make one more technical modification. We divide $\{1, 2, \ldots, i_0\}$ into groups $G_1, G_2, G_3, \ldots, G_r$, of contiguous subsets. We specify the groups later. Let $u_t = \text{Max}_{i \in G_t} v_i$ be the maximum

value of v_i within G_t. Define a new function $g(x)$ by $g(x) = u_t - 1$ for $x \in \cup_{i \in G_t}[\gamma_{i-1}, \gamma_i)$; see Figure 4.6. Since $g(x) \geq f(x)$

$$\int_0^{\gamma_{i_0}} f(x)\, dx \leq \int_0^{\gamma_{i_0}} g(x)\, dx. \tag{4.5}$$

We now assert (with $u_{r+1} = 1$):

$$\int_0^{\gamma_{i_0}} g(x)\, dx = \sum_{t=1}^{r} \pi(G_1 \cup G_2 \cup \ldots \cup G_t)(u_t - u_{t+1}). \tag{4.6}$$

This is just the statement that the area under $g(x)$ in the figure is exactly covered by the rectangles whose bottom sides are the dotted lines. We leave the formal proof of this to the reader. We now focus on proving that

$$\sum_{t=1}^{r} \pi(G_1 \cup G_2 \cup \ldots \cup G_t)(u_t - u_{t+1}) \leq \varepsilon/2, \tag{4.7}$$

for a sub-division into groups we specify which suffices by 4.3, 4.4, 4.5 and 4.6. While we start the proof of (4.7) with a technical observation (4.8), its proof will involve two nice ideas: the notion of probability flow and reckoning probability flow in two different ways. First, the technical observation: if $2\sum_{i \geq i_0+1}(1-v_i)\pi_i \leq \varepsilon$ then we would be done by (4.3). So assume now that $\sum_{i \geq i_0+1}(1 - v_i)\pi_i > \varepsilon/2$ from which it follows that $\sum_{i \geq i_0+1} \pi_i \geq \varepsilon/2$ and so, for any subset A of heavy nodes,

$$\text{Min}(\pi(A), \pi(\bar{A})) \geq \frac{\varepsilon}{2}\pi(A). \tag{4.8}$$

We now define the subsets. G_1 will be just $\{1\}$. In general, suppose $G_1, G_2, \ldots, G_{t-1}$ have already been defined. We start G_t at $i_t = 1+$ (end of G_{t-1}). Let $i_t = k$. We will define l, the last element of G_t to be the largest integer greater than or equal to k and at most i_0 so that

$$\sum_{j=k+1}^{l} \pi_j \leq \frac{\varepsilon \Phi \gamma_k}{4}.$$

In Lemma 4.6 which follows this theorem prove that for groups $G_1, G_2, \ldots, G_r, u_1.u_2, \ldots, u_r, u_{r+1}$ as above

$$\pi(G_1 \cup G_2 \cup \ldots G_r)(u_t - u_{t+1}) \leq \frac{8}{t\Phi\varepsilon}.$$

Now to prove (4.7), we only need an upper bound on r, the number of groups. If $G_t = \{k, k+1, \ldots, l\}$, with $l < i_0$, then by definition of l, we have $\gamma_{l+1} \geq (1 + \frac{\varepsilon\Phi}{2})\gamma_k$. So, $r \leq \ln_{1+(\varepsilon\Phi/2)}(1/\pi_1) + 2 \leq \ln(1/\pi_1)/(\varepsilon\Phi/2) + 2$. This completes the proof of (4.7) and the theorem. ∎

We complete the proof of Theorem 4.5 with the proof of Lemma 4.6. The notation in the lemma is that from the theorem.

Lemma 4.6 *Suppose groups $G_1, G_2, \ldots, G_r, u_1.u_2, \ldots, u_r, u_{r+1}$ are as above. Then,*

$$\pi(G_1 \cup G_2 \cup \ldots G_r)(u_t - u_{t+1}) \leq \frac{8}{t\Phi\varepsilon}.$$

Proof: This is the main lemma. The proof of the lemma uses a crucial idea of probability flows. We will use two ways of calculating the probability flow from heavy states to light states when we execute one step of the Markov chain starting at probabilities $\mathbf{a}$. The probability vector after that step is $\mathbf{a}P$. Now, $\mathbf{a} - \mathbf{a}P$ is the net loss of probability for each state due to the step.

Consider a particular group $G_t = \{k, k+1, \ldots, l\}$, say. First consider the case when $k < i_0$. Let $A = \{1, 2, \ldots, k\}$. The net loss of probability for each state from the set A in one step is $\sum_{i=1}^{k}(a_i - (\mathbf{a}P)_i)$ which is at most $\frac{2}{t}$ by the proof of Theorem 4.2.

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. For any $i < j$,

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

Thus, for any two states i and j, with i heavier than j, i.e., $i < j$, there is a non-negative net flow from i to j. (This is intuitively reasonable since it says that probability is flowing from heavy to light states.) Since $l \geq k$, the flow from A to $\{k+1, k+2, \ldots, l\}$ minus the flow from $\{k+1, k+2, \ldots, l\}$ to A is nonnegative. Since for $i \leq k$ and $j > l$, we have $v_i \geq v_k$ and $v_j \leq v_{l+1}$, 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,

$$(v_k - v_{l+1}) \sum_{\substack{i \leq k \\ j > l}} \pi_j p_{ji} \leq \frac{2}{t}. \tag{4.9}$$

Since

$$\sum_{i=1}^{k} \sum_{j=k+1}^{l} \pi_j p_{ji} \leq \sum_{j=k+1}^{l} \pi_j \leq \varepsilon\Phi\pi(A)/4$$

and by the definition of Φ, using (4.8)

$$\sum_{i \leq k < j} \pi_j p_{ji} \geq \Phi \text{Min}(\pi(A), \pi(\bar{A})) \geq \varepsilon\Phi\gamma_k/2,$$

we have, $\sum_{\substack{i \leq k \\ j > l}} \pi_j p_{ji} = \sum_{i \leq k < j} \pi_j p_{ji} - \sum_{i \leq k; j \leq l} \pi_j p_{ji} \geq \varepsilon\Phi\gamma_k/4$. Substituting this into the inequality (4.9) gives

$$v_k - v_{l+1} \leq \frac{8}{t\varepsilon\Phi\gamma_k}, \tag{4.10}$$

proving the lemma provided $k < i_0$. If $k = i_0$, the proof is similar but simpler. ∎

4.4.1 Using Normalized Conductance to Prove Convergence

We now apply Theorem 4.5 to some examples to illustrate how the normalized conductance bounds the rate of convergence. In each case we compute the mixing time for the uniform probability function on the vertices. Our first examples will be simple graphs. The graphs do not have rapid converge, but their simplicity helps illustrate how to bound the normalized conductance and hence the rate of convergence.

A 1-dimensional lattice

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, we have $p_{xy} = 1/2$ on all edges (x, y), and so the stationary distribution is a uniform $\frac{1}{n}$ over all vertices by Lemma 4.3. The set with minimum normalized conductance is the set S with probability $\pi(S) \leq \frac{1}{2}$ having the smallest ratio of probability mass exiting it, $\sum_{(x,y)\in(S,\bar{S})} \pi_x p_{xy}$, to probability mass inside it, $\pi(S)$. This set consists of the first $n/2$ vertices, for which the numerator is $\frac{1}{2n}$ and denominator is $\frac{1}{2}$. Thus,

$$\Phi(S) = \frac{1}{n}.$$

By Theorem 4.5, for ε a constant such as $1/100$, after $O(n^2 \log n/\epsilon^3)$ steps, $||\mathbf{a_t} - \boldsymbol{\pi}||_1 \leq 1/100$. This graph does not have rapid convergence. 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.

A 2-dimensional lattice

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 $1/4$. 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 $f_i p_{ij} = f_j p_{ji}$ and by Lemma 4.3, $\mathbf{f}$ is the stationary distribution. Consider any subset S consisting of at most half the states. If $|S| \geq \frac{n^2}{4}$, then the subset with the fewest edges leaving it consists of some number of columns plus perhaps one additional partial column. The number of edges leaving S is at least n. Thus

$$\sum_{i\in S} \sum_{j\in\bar{S}} \pi_i p_{ij} \geq \Omega\big(n\frac{1}{n^2}\big) = \Omega\left(\frac{1}{n}\right).$$

Since $|S| \geq \frac{n^2}{4}$, in this case

$$\Phi(S) \geq \Omega\left(\frac{1/n}{\min\left(\frac{S}{n^2}, \frac{\bar{S}}{n^2}\right)}\right) = \Omega\left(\frac{1}{n}\right).$$

If $|S| < \frac{n^2}{4}$, the subset S of a given size that has the minimum number of edges leaving consists of a square located at the lower left hand corner of the grid (Exercise 4.21). If

$|S|$ is not a perfect square then the right most column of S is short. Thus at least $2\sqrt{|S|}$ points in S are adjacent to points in $\bar{S}$. Each of these points contributes $\pi_i p_{ij} = \Omega(\frac{1}{n^2})$ to the flow$(S, \bar{S})$. Thus,

$$\sum_{i \in S} \sum_{j \in \bar{S}} \pi_i p_{ij} \geq \frac{c\sqrt{|S|}}{n^2}$$

and

$$\Phi(S) = \frac{\sum_{i \in S} \sum_{j \in \bar{S}} \pi_i p_{ij}}{\min\left(\pi(S), \pi(\bar{S})\right)} \geq \frac{c\sqrt{|S|}/n^2}{|S|/n^2} = \frac{c}{\sqrt{|S|}} = \Omega\left(\frac{1}{n}\right).$$

Thus, in either case, after $O(n^2 \ln n/\epsilon^3)$ steps, $|\mathbf{a}(\mathbf{t}) - \boldsymbol{\pi}|_1 \leq \epsilon$.

A lattice in d-dimensions

Next consider the $n \times n \times \cdots \times n$ lattice 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 lattice 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 4.22) that Φ is $\Omega(\frac{1}{dn})$. Since all π_i are equal to n^{-d}, the mixing time is $O(d^3 n^2 \ln n/\varepsilon^3)$, which is polynomially bounded in n and d.

The d-dimensional lattice is related to the Metropolis-Hastings algorithm and Gibbs sampling although in those constructions there is a nonuniform probability distribution at the vertices. However, the d-dimension lattice case suggests why the Metropolis-Hastings and Gibbs sampling constructions might converge fast.

A clique

Consider an n vertex clique with a self loop at each vertex. For each edge, $p_{xy} = \frac{1}{n}$ and thus for each vertex, $\pi_x = \frac{1}{n}$. Let S be a subset of the vertices. Then

$$\sum_{x \in S} \pi_x = \frac{|S|}{n}.$$

$$\sum_{(x,y) \in (S, \bar{S})} \pi_x p_{xy} = \pi_x p_{xy} |S||\overline{S}| = \frac{1}{n^2} |S||\overline{S}|$$

and

$$\Phi(S) = \frac{\sum_{(x,y) \in (S, \bar{S})} \pi_x p_{xy}}{\min(\sum_{x \in S} \pi_x, \sum_{x \in \bar{S}} \pi_x)} = \frac{\frac{1}{n^2}|S||\overline{S}|}{\min(\frac{1}{n}|S|, \frac{1}{n}|\overline{S}|)} = \frac{1}{n} \max(|S|, |\overline{S}|) = \frac{1}{2}.$$

This gives a bound on the ε-mixing time of

$$O\left(\frac{\ln \frac{1}{\pi_{\min}}}{\Phi^2 \epsilon^3}\right) = O\left(\frac{\ln n}{\epsilon^3}\right).$$

However, a walker on the clique starting from any probability distribution will in one step be exactly at the stationary probability distribution.

A connected undirected graph

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. That is, if the degree of vertex x is d_x and the number of edges in the graph is m, then $\pi_x = \frac{d_x}{2m}$. Notice that for any edge (x, y) we have

$$\pi_x p_{xy} = \left(\frac{d_x}{2m}\right)\left(\frac{1}{d_x}\right) = \frac{1}{2m}.$$

Therefore, for any S, the total conductance of edges out of S is at least $\frac{1}{2m}$, and so Φ is at least $\frac{1}{m}$. Since $\pi_{\min} \geq \frac{1}{2m} \geq \frac{1}{n^2}$, $\ln \frac{1}{\pi_{min}} = O(\ln n)$. Thus, the mixing time is $O(m^2 \ln n/\varepsilon^3) = O(n^4 \ln n/\varepsilon^3)$.

The Gaussian distribution on the interval [-1,1]

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 $\boldsymbol{\pi}(S) \leq \frac{1}{2}$. First consider the case when S is an interval $[k\delta, 1]$ for $k \geq 2$. 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 total conductance of edges out of S is

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

Using $2 \leq k \leq 1/\delta$, $\alpha \geq 1$, and $\pi(\bar{S}) \leq 1$,

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

For the grid size less than the variance of the Gaussian distribution, $\delta < \frac{1}{\alpha}$, we have $\alpha\delta < 1$, so $e^{-O(\alpha\delta)} = \Omega(1)$, thus, $\Phi(S) \geq \Omega(\alpha\delta)$. 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 4.5 in $\Omega(1/\alpha^2\delta^3\varepsilon^3)$ steps, a walk gets within ε of the steady state distribution.

In the uniform probability case the ϵ-mixing time is bounded by $n^2 \log n$. For comparison, in the Gaussian case set $\delta = 1/n$ and $\alpha = 1/3$. This gives an ϵ-mixing time bound of n^3. In the Gaussian case with the entire initial probability on the first vertex, the chain begins to converge faster to the stationary probability than the uniform distribution case since the chain favors higher degree vertices. However, ultimately the distribution must reach the lower probability vertices on the other side of the Gaussian's maximum and here the chain is slower since it favors not leaving the higher probability vertices.

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

4.5 Electrical Networks and Random Walks

In the next few sections, we study the relationship between electrical networks and random walks on undirected graphs. The graphs have nonnegative weights on each edge. A step is executed by picking a random edge from the current vertex with probability proportional to the edge's weight and traversing the edge.

An electrical network is a connected, undirected graph in which each edge (x, y) 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 than resistance. Associated with an electrical network is a random walk on the underlying graph defined by assigning a probability $p_{xy} = c_{xy}/c_x$ to the edge (x, y) incident to the vertex x, where the normalizing constant c_x equals $\sum_y c_{xy}$. Note that although c_{xy} equals c_{yx}, the probabilities p_{xy} and p_{yx} may not be equal due to the normalization required to make the probabilities at each vertex sum to one. We shall soon see that there is a relationship between current flowing in an electrical

network and a random walk on the underlying graph.

Since we assume that the undirected graph is connected, by Theorem 4.2 there is a unique stationary probability distribution.The stationary probability distribution is $\boldsymbol{\pi}$ where $\pi_x = \frac{c_x}{c_0}$ with $c_0 = \sum_x c_x$. To see this, for all x and y

$$\pi_x p_{xy} = \frac{c_x}{c_0}\frac{c_{xy}}{c_x} = \frac{c_y}{c_0}\frac{c_{yx}}{c_y} = \pi_y p_{yx}$$

and hence by Lemma 4.3, $\boldsymbol{\pi}$ is the unique stationary probability.

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 a function whose value at the boundary vertices is fixed to some boundary condition, and whose value at any interior vertex x is a weighted average of its values at all the adjacent vertices y, with weights p_{xy} satisfying $\sum_y p_{xy} = 1$ for each x. Thus, if at every interior vertex x for some set of weights p_{xy} satisfying $\sum_y p_{xy} = 1$, $g_x = \sum_y g_y p_{xy}$, then g is an harmonic function.

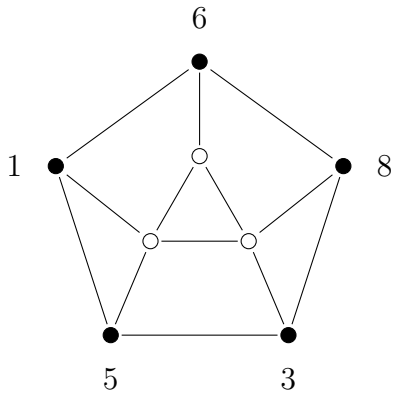
Example: Convert an electrical network with conductances c_{xy} to a weighted, undirected graph with probabilities p_{xy}. Let $\mathbf{f}$ be a function satisfying $\mathbf{f}P = \mathbf{f}$ where P is the matrix of probabilities. It follows that the function $g_x = \frac{f_x}{c_x}$ is harmonic.

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

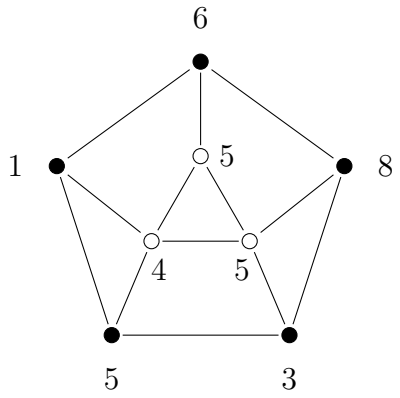
∎

A harmonic function on a connected graph takes on its maximum and minimum on the boundary. This is easy to see for the following reason. Suppose the maximum does not occur on the boundary. Let S be the set of 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 $y \in \bar{S}$. The value of the function at x is the weighted 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



Graph with boundary vertices dark and boundary conditions specified.



Values of harmonic function satisfying boundary conditions where the edge weights at each vertex are equal

Figure 4.7: Graph illustrating an harmonic function.

solutions is itself harmonic. Since $h(x) = f(x) - g(x)$ is harmonic and has value zero on the boundary, by the min and max principles it has value zero everywhere. Thus $f(x) = g(x)$.

The analogy between electrical networks and random walks

There are important connections between electrical networks and random walks on undirected graphs. Choose two vertices a and b. Attach a voltage source between a and b so that the voltage v_a equals one volt and the voltage v_b equals zero. 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. What we will show below is the following. Having fixed the voltages at the vertices a and b, the voltage at an arbitrary vertex x equals the probability that a random walk that starts at x will reach a before it reaches b. We will also show there is a related probabilistic interpretation of current as well.

Probabilistic interpretation of voltages

Before relating voltages and probabilities, 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 Kirchhoff'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 $\sum_y c_{xy} = c_x$ and that $p_{xy} = \frac{c_{xy}}{c_x}$, yields $v_x c_x = \sum_y v_y p_{xy} c_x$. Hence, $v_x = \sum_y v_y p_{xy}$. Thus, the voltage at each vertex x is a weighted average of the voltages at the adjacent vertices. Hence the voltages form a harmonic function with $\{a, b\}$ as the boundary.

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 in the first step 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 at a and b.. 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 a 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 $u_b = 0$. Consider a node x not equal to a or b. Every time the walk visits x, it must have come from some neighbor y. Thus, the expected number of visits to x before reaching b is the sum over all neighbors y of the expected number of visits u_y to y before reaching b times the probability p_{yx} of going from y to x. That is,

$$u_x = \sum_y u_y p_{yx}.$$

Since $c_x p_{xy} = c_y p_{yx}$

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

and hence $\frac{u_x}{c_x} = \sum_y \frac{u_y}{c_y} p_{xy}$. It follows that $\frac{u_x}{c_x}$ is harmonic with a and b as the boundary where the boundary conditions are $u_b = 0$ and u_a equals some fixed value. 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 $\frac{u_a}{c_a}$. Now $\frac{u_x}{c_x}$ and v_x satisfy the same boundary conditions and thus are the same harmonic function. Let the current into a correspond to one walk. Note that if the 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. This implies that 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 $\frac{c_{eff}}{c_a}$. For convenience, assume that a and b are not adjacent. A slight modification of the argument suffices for the case when a and b are adjacent.

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

Since $v_a = 1$,

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

For each y adjacent to the vertex a, p_{ay} is the probability of the walk going from vertex a to vertex y. Earlier we showed that v_y is the probability of a walk starting at y going to a before reaching b. Thus, $\sum_y p_{ay} v_y$ is the probability of a walk starting at a returning to a before reaching b and $1 - \sum_y p_{ay} v_y$ 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 connected graph, the escape probability will always be nonzero. 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.

4.6 Random Walks on Undirected Graphs with Unit Edge Weights

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. 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 given starting probability distribution.

One interesting fact is that adding edges to a graph may either increase or decrease h_{xy} depending on the particular situation. Adding 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 path of n vertices is $\Theta(n^2)$.

Lemma 4.7 *The expected time for a random walk starting at one end of a path of n vertices to reach the other end is $\Theta(n^2)$.*

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

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

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, you need to first reach 2, then from 2 (eventually) reach 3, then from 3 (eventually) reach 4, and so on. Thus by linearity of expectation,

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

■

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 $2 \leq i \leq n-1$.

Lemma 4.8 *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*

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

Proof: Now $t\,(n) = 1$ 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 $t\,(n-1) = 2$. For $3 \leq i < n-1$, $t\,(i) = \frac{1}{2}\left[t\,(i-1) + t\,(i+1)\right]$ and $t\,(1)$ and $t\,(2)$ satisfy $t\,(1) = \frac{1}{2}t\,(2) + 1$ and $t\,(2) = t\,(1) + \frac{1}{2}t\,(3)$. Solving for $t(i+1)$ for $3 \leq i < n-1$ yields

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

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

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

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

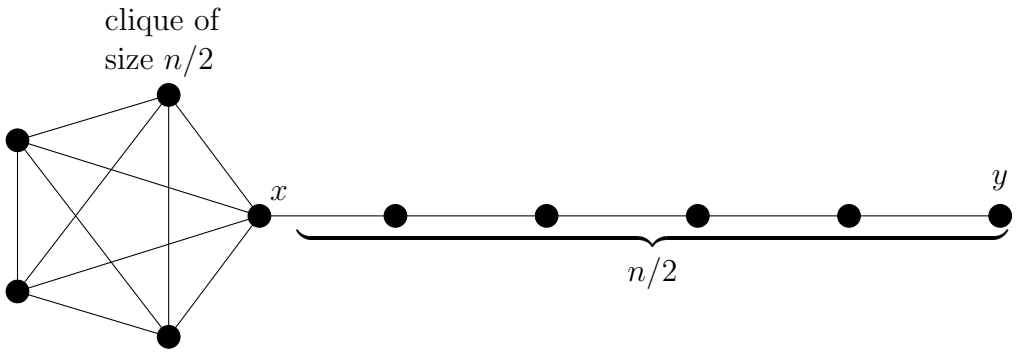


Figure 4.8: Illustration that adding edges to a graph can either increase or decrease hitting time.

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 4.8 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 the figure. In the first graph consisting of a single path of length n, $h_{xy} = \Theta(n^2)$. In the second graph consisting of a clique of size $n/2$ along with a path of length $n/2$, $h_{xy} = \Theta(n^3)$.To see this latter statement, note that starting at x, the walk will go down the path towards y and return to x for $n/2-1$ times on average before reaching y for the first time, by Lemma 4.8. Each time the walk in the path returns to x, with probability $(n/2-1)/(n/2)$ it enters the clique and thus on average enters the clique $\Theta(n)$ times before starting down the path again. Each time it enters the clique, it spends $\Theta(n)$ time in the clique before returning to x. It then reenters the clique $\Theta(n)$ times before starting down the path to y. Thus, each time the walk returns to x from the path it spends $\Theta(n^2)$ time in the clique before starting down the path 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, $h_{xy} = \Theta(n)$. 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 4.8, 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 $\Theta(n^2)$.

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. So commute$(x, y) = h_{xy} + h_{yx}$. Think of going from home to office and returning home. Note that commute time is symmetric. We now relate the commute time to an electrical quantity, the effective resistance. The *effective resistance* between two vertices x and y in an electrical network is the voltage difference

between x and y when one unit of current is inserted at vertex x and withdrawn from vertex y.

Theorem 4.9 *Given a connected, 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 (note: for this to be legal, the graph must be connected). Let v_{ij} be the voltage difference from i to j. The current into i divides into the d_i resistors at vertex 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 resistor. The sum of the currents out of i through the resistors must equal d_i, the current injected into i.

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

Solving for v_{ij}

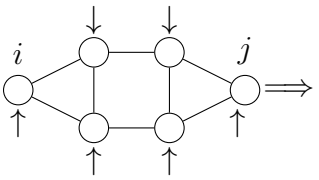
$$v_{ij} = 1 + \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i} v_{kj} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i}(1 + v_{kj}). \tag{4.11}$$

Now the hitting 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

$$h_{ij} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \tfrac{1}{d_i}(1 + h_{kj}). \tag{4.12}$$

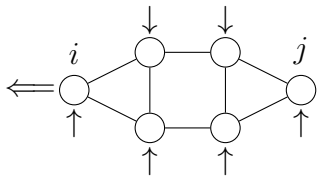
Subtracting (4.12) from (4.11), gives $v_{ij} - h_{ij} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i}(v_{kj} - h_{kj})$. Thus, the function $v_{ij} - h_{ij}$ is harmonic. Designate vertex j as the only boundary vertex. The value of $v_{ij} - h_{ij}$ at $i = j$, namely $v_{jj} - h_{jj}$, 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 of Theorem 4.9, note that $h_{ij} = v_{ij}$ is the voltage from i to j when currents are inserted at all vertices in the graph and extracted at vertex j. If the current is extracted from i instead of j, then the voltages change and $v_{ji} = h_{ji}$ in the new



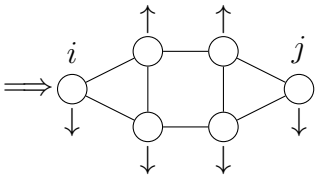
Insert current at each vertex
equal to degree of the vertex.
Extract $2m$ at vertex j, $v_{ij} = h_{ij}$.

(a)



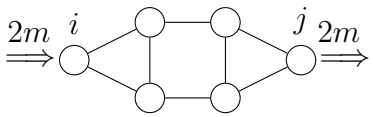
Extract current from i instead of j.
For new voltages $v_{ji} = h_{ji}$.

(b)



Reverse currents in (b).
For new voltages $-v_{ji} = h_{ji.}$
Since $-v_{ji} = v_{ij}, h_{ji} = v_{ij}$.

(c)



Superpose currents in (a) and (c).
$2mr_{ij} = v_{ij} = h_{ij} + h_{ji} = commute(i, j)$.

(d)

Figure 4.9: Illustration of proof that commute$(x, y) = 2mr_{xy}$ where m is the number of edges in the undirected graph and r_{xy} is the effective resistance between x and y.

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 vertex equal to the degree of the vertex 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}, which is the sum of the other two v_{ij}'s, is $v_{ij} = h_{ij} + h_{ji}$. All currents into or out of the network cancel except the $2m$ amps injected at i and withdrawn at j. Thus, $2mr_{ij} = v_{ij} = h_{ij} + h_{ji} =$ commute(i, j) or commute$(i, j) = 2mr_{ij}$ where r_{ij} is the effective resistance from i to j. ∎

The following corollary follows from Theorem 4.9 since the effective resistance r_{uv} is less than or equal to one when u and v are connected by an edge.

Corollary 4.10 *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.*

Proof: If x and y are connected by an edge, then the effective resistance r_{xy} is less than or equal to one. ∎

Corollary 4.11 *For vertices x and y in an n vertex graph, the commute time, commute(x, y), is less than or equal to n^3.*

Proof: By Theorem 4.9 the commute time is given by the formula 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. Since the resistance can not be greater than that of any path from x to y, $r_{xy} \leq n$. Since the number of edges is at most $\binom{n}{2}$

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

■

While adding edges into a graph can never increase the effective resistance between two given nodes x and y, it may increase or decrease the commute time. To see this consider three graphs: the graph consisting of a chain of n vertices, the graph of Figure 4.8, and the clique on n vertices.

Cover time

The *cover time*, 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 cover(x) when G is understood. The cover time of an undirected graph G, denoted 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 4.8 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 4.8 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 $\Theta(n \log n)$ since this 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 4.8, the cover time is $\Theta(n^3)$ since one takes the maximum over all start states and cover$(x, G) = \Theta(n^3)$ where x is the vertex of attachment.

Theorem 4.12 *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 $4m(n-1)$.*

Proof: Consider a depth first search of the graph G starting from some vertex z and let T be the resulting depth first search spanning tree of G. The depth first search 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. Note that each edge in T is traversed twice, once in each direction.

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

If (x, y) is an edge in T, then x and y are adjacent and thus Corollary 4.10 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, $\text{cover}(z) \leq 4m(n-1)$. This holds for all starting vertices z. Thus, $\text{cover}(G) \leq 4m(n-1)$. ∎

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 $r_{\mathit{eff}}(G)$ of a graph G by $r_{\mathit{eff}}(G) = \max_{x,y}(r_{xy})$.

Theorem 4.13 *Let G be an undirected graph with m edges. Then the cover time for G is bounded by the following inequality*

$$m\, r_{\mathit{eff}}(G) \leq \mathit{cover}(G) \leq 6e\, m\, r_{\mathit{eff}}(G) \ln n + n$$

where $e \approx 2.718$ is Euler's constant and $r_{\mathit{eff}}(G)$ is the resistance of G.

Proof: By definition $r_{\mathit{eff}}(G) = \max_{x,y}(r_{xy})$. Let u and v be the vertices of G for which r_{xy} is maximum. Then $r_{\mathit{eff}}(G) = r_{uv}$. By Theorem 4.9, $\text{commute}(u, v) = 2mr_{uv}$. Hence $mr_{uv} = \frac{1}{2}\text{commute}(u, v)$. Note that $\frac{1}{2}\text{commute}(u, v)$ is the average of h_{uv} and h_{vu}, which is clearly less than or equal to $\max(h_{uv}, h_{vu})$. Finally, $\max(h_{uv}, h_{vu})$ is less than or equal to $\max(\text{cover}(u, G), \text{cover}(v, G))$ which is clearly less than the cover time of G. Putting these facts together gives the first inequality in the theorem.

$$m\, r_{\mathit{eff}}(G) = mr_{uv} = \tfrac{1}{2}\text{commute}(u, v) \leq \max(h_{uv}, h_{vu}) \leq \text{cover}(G)$$

For the second inequality in the theorem, by Theorem 4.9, for any x and y, $\text{commute}(x, y)$ equals $2mr_{xy}$ which is less than or equal to $2m\, r_{\mathit{eff}}(G)$, implying $h_{xy} \leq 2m\, r_{\mathit{eff}}(G)$. By the Markov inequality, since the expected time to reach y starting at any x is less than $2m\, r_{\mathit{eff}}(G)$, the probability that y is not reached from x in $2m\, r_{\mathit{eff}}(G)e$ steps is at most $\frac{1}{e}$. Thus, the probability that a vertex y has not been reached in $6e\, m\, r_{\mathit{eff}}(G) \log n$ steps is at most $\frac{1}{e}^{3 \ln n} = \frac{1}{n^3}$ because a random walk of length $6e\, mr_{\mathit{eff}}(G) \log n$ is a sequence of $3 \log n$ random walks, each of length $2emr_{\mathit{eff}}(G)$ and each possibly starting from different

vertices. Suppose after a walk of $6em\, r_{eff}(G) \log n$ steps, vertices $v_1, v_2, \ldots, v_l$ had not been reached. Walk until v_1 is reached, then v_2, etc. By Corollary 4.11 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 at most n. More precisely,

$$\begin{aligned}\text{cover}(G) &\leq 6em\, r_{eff}(G) \log n + \sum_v \text{Prob}\left(v \text{ was not visited in the first } 6em\, r_{eff}(G) \text{ steps}\right) n^3 \\ &\leq 6em\, r_{eff}(G) \log n + \sum_v \frac{1}{n^3} n^3 \leq 6em\, r_{eff}(G) + n.\end{aligned}$$

■

4.7 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.

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 resistor. 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 conductances at the origin. In a d-dimensional lattice, $c_a = 2d$ assuming that the resistors have value one. For the d-dimensional lattice

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

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

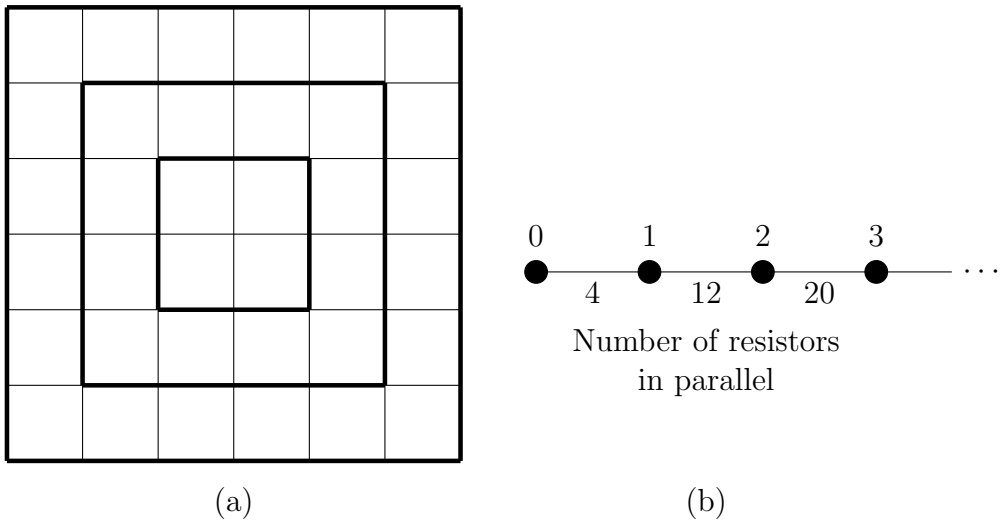


Figure 4.10: 2-dimensional lattice along with the linear network resulting from shorting resistors on the concentric squares about the origin.

dimensional lattice will return to the origin with probability one. Note, however, that the expected time to return to the origin having taken one step away, which is equal to commute$(1, 0)$, is infinite (Theorem 4.9.

Two dimensions

For the 2-dimensional lattice, consider a larger and larger square about the origin for the boundary as shown in Figure 4.10a 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 4.10b. As the paths get longer, the number of resistors in parallel also increases. The resistance between vertex i and $i+1$ is really $4(2i+1)$ unit resistors in parallel. The effective resistance of $4(2i+1)$ resistors in parallel is $1/4(2i+1)$. 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 and hence 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 there are a sufficient number of paths that r_{eff} remains finite and thus there is a nonzero escape probability. We will prove this now. First note that shorting any edge decreases the resistance, so

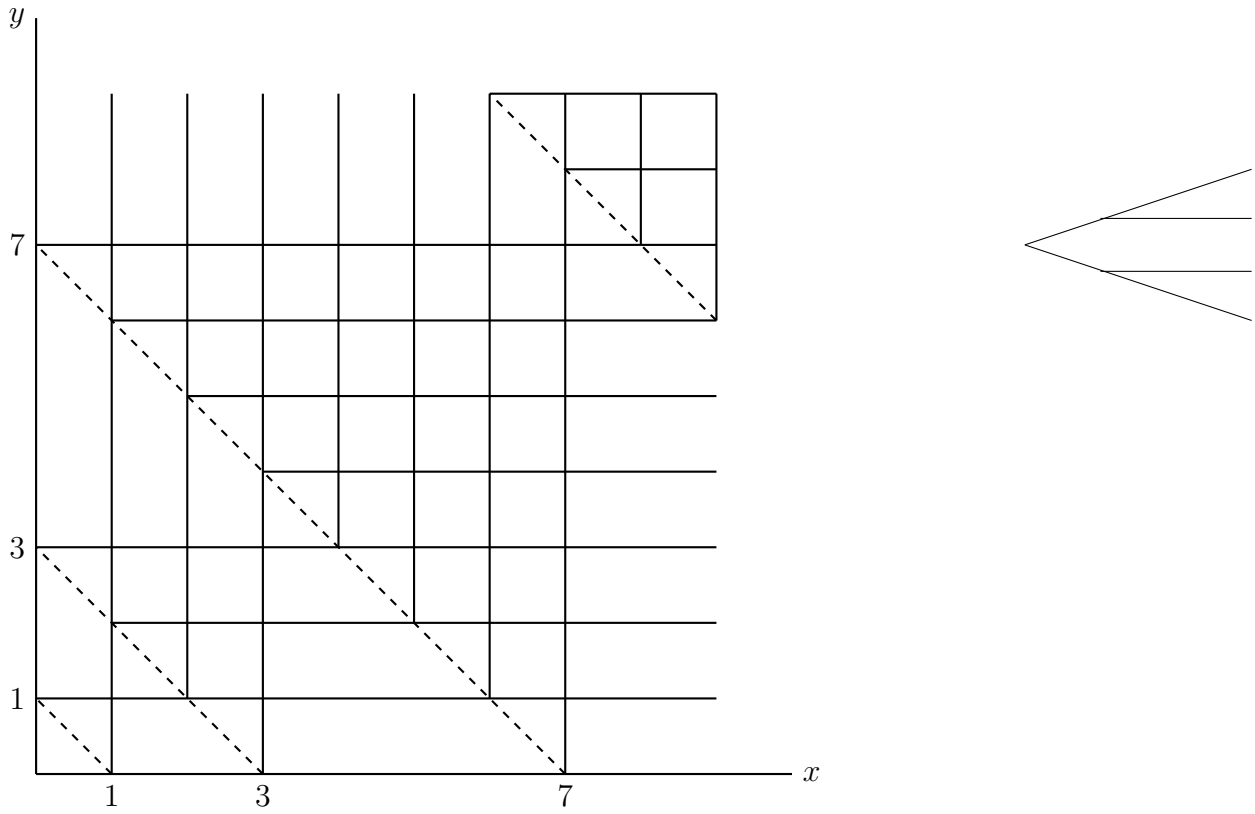


Figure 4.11: Paths in a 2-dimensional lattice obtained from the 3-dimensional construction applied in 2-dimensions.

we do not use shorting in this proof, since we seek to prove an upper bound on the resistance. Instead we remove some edges, which increases their resistance to infinity and hence increases the effective resistance, giving an upper bound. To simplify things we consider walks on a quadrant rather than the full grid. The resistance to infinity derived from only the quadrant is an upper bound on the resistance of the full grid.

The construction used in three dimensions is easier to explain first in two dimensions, see Figure 4.11. 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

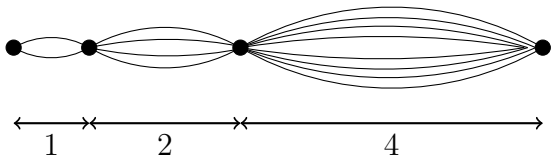


Figure 4.12: Paths obtained from 2-dimensional lattice. Distances between splits double as do the number of parallel paths.

of paths in parallel doubles. See Figure 4.12. 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. It is important to check that the paths are edge-disjoint and so the tree is a subgraph of the lattice. Going to a subgraph is equivalent to deleting edges which increases the resistance. That is why the resistance of the lattice is less than that of the tree. Thus, in three dimensions the escape probability is nonzero. The upper bound on r_{eff} gives the lower bound

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

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

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

This gives

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

4.8 The Web as a Markov Chain

A modern application of random walks on directed graphs comes from trying to establish the importance of pages on the World Wide Web. Search Engines output an ordered

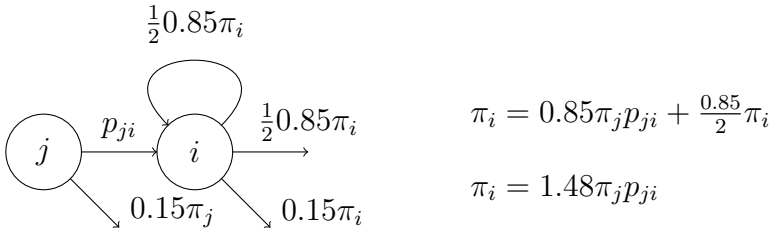


Figure 4.13: Impact on pagerank of adding a self loop

list of webpages in response to each search query. To do this, they have to solve two problems at query time: (i) find the set of all webpages containing the query term(s) and (ii) rank the webpages and display them (or the top subset of them) in ranked order. (i) is done by maintaining a "reverse index" which we do not discuss here. (ii) cannot be done at query time since this would make the response too slow. So Search Engines rank the entire set of webpages (in the billions) "off-line" and use that single ranking for all queries. At query time, the webpages containing the query terms(s) are displayed in this ranked order.

One way to do this ranking would be to take a random walk on the web viewed as a directed graph (which we call the web graph) with an edge corresponding to each hypertext link and rank pages according to their stationary probability. Hypertext links are one-way and the web graph may not be strongly connected. Indeed, for a node at the "bottom" level there may be no out-edges. When the walk encounters this vertex the walk disappears. Another difficulty is that a vertex 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 vertex selected uniformly at random in the entire graph; with probability $1 - r$ select an out-edge at random from the current node and follow it. If a vertex has no out edges, the value of r for that vertex is set to one. This makes the graph strongly connected so that the stationary probabilities exist.

Pagerank

The pagerank of a vertex in a directed graph is the stationary probability of the vertex, where we assume a positive restart probability of say $r = 0.15$. The restart ensures that the graph is strongly connected. The pagerank of a page is the frequency with which the page will be visited over a long period of time. If the pagerank 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 vertex i with a single edge in from vertex 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 pagerank 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

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

Solving for π_i yields

$$\pi_i = \frac{0.85k + 0.85}{0.15k + 1}\pi_j p_{ji}$$

which for $k = 1$ is $\pi_i = 1.48\pi_j p_{ji}$ and in the limit as $k \to \infty$ is $\pi_i = 5.67\pi_j p_{ji}$. Adding a single loop only increases pagerank by a factor of 1.74.

Relation to Hitting time

Recall the definition of hitting time h_{xy}, which for two states x and y is the expected time to reach y starting from x. Here, we deal with h_y, the average time to hit y, starting at a random node. Namely, $h_y = \frac{1}{n}\sum_x h_{xy}$, where the sum is taken over all n nodes x. Hitting time h_y is closely related to return time and thus to the reciprocal of page rank. Return time is clearly less than the expected time until a restart plus hitting time. With r as the restart value, this gives:

$$\text{Return time to } y \leq \frac{1}{r} + h_y.$$

In the other direction, the fastest one could return would be if there were only paths of length two (assume we remove all self-loops). A path of length two 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 $2\,(1-r)^2 + (2-r)\,r \times$ (hitting time). Combining these two bounds yields

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

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

Spam

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

The pagerank of v is the stationary probability for vertex v with random restarts. If we delete all existing edges out of v, create a new vertex u and edges (v, u) and (u, v), then the pagerank 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 pagerank would be to create a star consisting of the vertex v at its center along with a large set of new vertices each with a directed edge to v. These new vertices will sometimes be chosen as the target of the random restart and hence the vertices 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 pagerank but does not effect hitting time. One can negate the impact on pagerank of someone capturing the random walk by increasing the frequency of random restarts. The second technique of creating a star increases pagerank due to random restarts and decreases hitting time.

One can check if the pagerank is high and hitting time is low in which case the pagerank is likely to have been artificially inflated by the page capturing the walk with short cycles.

Personalized pagerank

In computing pagerank, 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 vertex selected uniformly at random. In personalized pagerank, instead of selecting a vertex uniformly at random, one selects a vertex according to a personalized probability distribution. Often the distribution has probability one for a single vertex and whenever the walk restarts it restarts at that vertex. Note that this may make the graph disconnected.

Algorithm for computing personalized pagerank

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

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

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

or

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

Thus, in principle, $\mathbf{p}$ can be found by computing the inverse of $[I-(1-\alpha)A]^{-1}$. 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 3 that the process converges to the solution $\mathbf{p}$.

For the personalized pagerank, 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.

4.9 Bibliographic Notes

The material on the analogy between random walks on undirected graphs and electrical networks is from [DS84] as is the material on random walks in Euclidean space. Addi-

tional material on Markov chains can be found in [MR95b], [MU05], and [per10]. For material on Markov Chain Monte Carlo methods see [Jer98] and [Liu01].

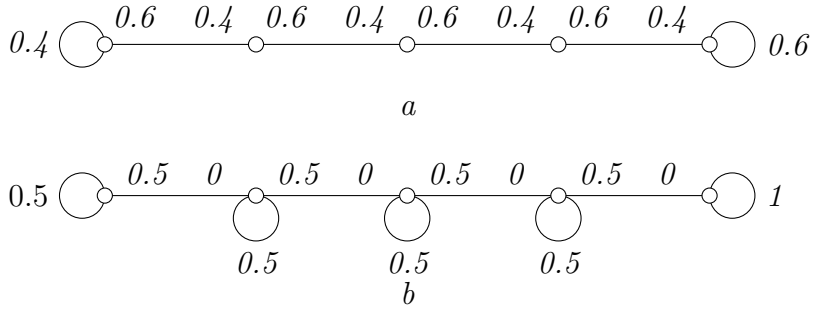
The use of normalized conductance to prove convergence of Markov Chains is by Sinclair and Jerrum, [SJ89] and Alon [Alo86]. A polynomial time bounded Markov chain based method for estimating the volume of convex sets was developed by Dyer, Frieze and Kannan [DFK91].

4.10 Exercises

Exercise 4.1 *The Fundamental Theorem of Markov chains says that for a connected Markov chain, the long-term average distribution* $\mathbf{a}(\mathbf{t})$ *converges to a stationary distribution. Does the* t *step distribution* $\mathbf{p}(\mathbf{t})$ *also converge for every connected Markov Chain? Consider the following examples: (i) A two-state chain with* $p_{12} = p_{21} = 1$. *(ii) A three state chain with* $p_{12} = p_{23} = p_{31} = 1$ *and the other* $p_{ij} = 0$. *Generalize these examples to produce Markov Chains with many states.*

Exercise 4.2 *Does* $\lim_{t\to\infty} a(t) - a(t+1) = 0$ *imply that* $a(t)$ *converges to some value? Hint: consider the average cumulative sum of the digits in the sequence* $10^2 1^4 0^8 1^{16} \cdots$

Exercise 4.3 *What is the stationary probability for the following networks.*



Exercise 4.4 *A Markov chain is said to be symmetric if for all* i *and* j, $p_{ij} = p_{ji}$. *What is the stationary distribution of a connected symmetric chain? Prove your answer.*

Exercise 4.5 *Prove* $|\mathbf{p} - \mathbf{q}|_1 = 2\sum_i (p_i - q_i)^+$ *for probability distributions* $\mathbf{p}$ *and* $\mathbf{q}$, *(Proposition 4.4).*

Exercise 4.6 *Let* $p(\mathbf{x})$, *where* $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ $x_i \in \{0, 1\}$, *be a multivariate probability distribution. For* $d = 100$, *how would you estimate the marginal distribution*

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

Exercise 4.7 *Using the Metropolis-Hasting Algorithm create a Markov chain whose stationary probability is that given in the following table. Use the* 3×3 *lattice for the underlying graph.*

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 4.8 *Using Gibbs sampling create a* 4×4 *lattice where vertices in rows and columns are cliques whose stationary probability is that given in the following table.*

x/y	1	2	3	4
1	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$
2	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{32}$
3	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{32}$
4	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$

Note by symmetry there are only three types of vertices and only two types of rows or columns.

Exercise 4.9 *How would you integrate a high dimensional multivariate polynomial distribution over some convex region?*

Exercise 4.10 *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 stationary distribution. What happens to the convergence time in this modification?*

Exercise 4.11 *Let* $\mathbf{p}$ *be a probability vector (nonnegative components adding up to 1) on the vertices of a connected graph which is sufficiently large that it cannot be stored in a computer. 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 stationary probability vector is* $\mathbf{p}$*. Is a random walk an efficient way to sample according to a probability distribution that is close to* $\mathbf{p}$*? Think, for example, of the graph* G *being the* n*-dimensional hypercube with* 2^n *vertices, and* $\mathbf{p}$ *as the uniform distribution over those vertices.*

Exercise 4.12 *Construct the edge probability for a three state Markov chain where each pair of states is connected by an undirected edge so that the stationary probability is* $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$*. Repeat adding a self loop with probability* $\frac{1}{2}$ *to the vertex with probability* $\frac{1}{2}$*.*

Exercise 4.13 *Consider a three state Markov chain with stationary 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 vertices. For each edge and each direction what is the expected probability that we would actually make a move along the edge?*

Exercise 4.14 *Consider a distribution* $\mathbf{p}$ *over* $\{0,1\}^2$ *with* $p(00) = p(11) = \frac{1}{2}$ *and* $p(01) = p(10) = 0$*. Give a connected graph on* $\{0,1\}^2$ *that would be bad for running Metropolis-Hastings and a graph that would be good for running Metropolis-Hastings. What would be the problem with Gibbs sampling?*

Exercise 4.15 *Consider* $p(\mathbf{x})$ *where* $\mathbf{x} \in \{0,1\}^{100}$ *such that* $p(\mathbf{0}) = \frac{1}{2}$ *and* $p(\mathbf{x}) = \frac{1/2}{(2^{100}-1)}$ *for* $\mathbf{x} \neq \mathbf{0}$*. How does Gibbs sampling behave?*

Exercise 4.16 *Given a connected graph G and an integer k how would you generate connected subgraphs of G with k vertices with probability proportional to the number of edges in the subgraph? A subgraph of G does not need to have all edges of G that join vertices of the subgraph. The probabilities need not be exactly proportional to the number of edges and you are not expected to prove your algorithm for this problem.*

Exercise 4.17 *Suppose one wishes to generate uniformly at random a regular, degree three, undirected, not necessarily connected multi-graph with 1,000 vertices. A multi-graph may have multiple edges between a pair of vertices and self loops. One decides to do this by a Markov Chain Monte Carlo technique. In particular, consider a (very large) network where each vertex corresponds to a regular degree three, 1,000 vertex multi-graph. For edges, say that the vertices corresponding to two graphs are connected by an edge if one graph can be obtained from the other by a flip of a pair of edges. In a flip, a pair of edges (a, b) and (c, d) are replaced by (a, c) and (b, d).*

1. *Prove that the network whose vertices correspond to the desired graphs is connected. That is, for any two 1000-vertex degree-3 multigraphs, it is possible to walk from one to the other in this network.*

2. *Prove that the stationary probability of the random walk is uniform over all vertices.*

3. *Give an upper bound on the diameter of the network.*

4. *How would you modify the process if you wanted to uniformly generate connected degree three multi-graphs?*

In order to use a random walk to generate the graphs in a reasonable amount of time, the random walk must rapidly converge to the stationary probability. Proving this is beyond the material in this book.

Exercise 4.18 *Construct, program, and execute an algorithm to estimate the volume of a unit radius sphere in 20 dimensions by carrying out a random walk on a 20 dimensional grid with 0.1 spacing.*

Exercise 4.19 *What is the mixing time for the undirected graphs*

1. *Two cliques connected by a single edge?*

2. *A graph consisting of an n vertex clique plus one additional vertex connected to one vertex in the clique.*

Exercise 4.20 *What is the mixing time for*

1. *$G(n, p)$ with $p = \frac{\log n}{n}$?*

2. *A circle with n vertices where at each vertex an edge has been added to another vertex chosen at random. On average each vertex will have degree four, two circle edges, and an edge from that vertex to a vertex chosen at random, and possible some edges that are the ends of the random edges from other vertices.*

Exercise 4.21 *Find the ϵ-mixing time for a 2-dimensional lattice with n vertices in each coordinate direction with a uniform probability distribution. To do this solve the following problems.*

1. *The minimum number of edges leaving a set S of size greater than or equal to $n^2/4$ is n.*
2. *The minimum number of edges leaving a set S of size less than or equal to $n^2/4$ is $\lfloor\sqrt{S}\rfloor$.*
3. *Compute $\Phi(S)$*
4. *Compute Φ*
5. *Computer the ϵ-mixing time*

Exercise 4.22 *Find the ϵ-mixing time for a d-dimensional lattice with n vertices in each coordinate direction with a uniform probability distribution. To do this, solve the following problems.*

1. *Select a direction say x_1 and push all elements of S in each column perpendicular to $x_1 = 0$ as close to $x_1 = 0$ as possible. Prove that the number of edges leaving S is at least as large as the number leaving the modified version of S.*
2. *Repeat step one for each direction. Argue that for a direction say x_1, as x_1 gets larger a set in the perpendicular plane is contained in the previous set.*
3. *Optimize the arrangements of elements in the plane $x_1 = 0$ and move elements from farthest out plane in to make all planes the same shape as $x_1 = 0$ except for some leftover elements of S in the last plane. Argue that this does not increase the number of edges out.*
4. *What configurations might we end up with?*
5. *Argue that for a given size, S has at least as many edges as the modified version of S.*
6. *What is $\Phi(S)$ for a modified form S?*
7. *What is Φ for a d-dimensional lattice?*
8. *What is the ϵ-mixing time?*

Exercise 4.23

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

2. *Let q_x be the stationary probability of vertex x in a random walk on an undirected graph where all edges at a vertex are equally likely and let d_x be the degree of vertex x. Show that $\frac{q_x}{d_x}$ is a harmonic function.*

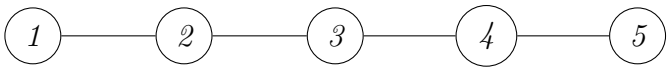
3. *If there are multiple harmonic functions when there are no boundary conditions, why is the stationary probability of a random walk on an undirected graph unique?*

4. *What is the stationary probability of a random walk on an undirected graph?*

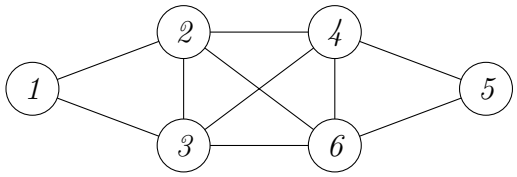
Exercise 4.24 *In Section 4.5, given an electrical network, we define an associated Markov chain such that voltages and currents in the electrical network corresponded to properties of the Markov chain. Can we go in the reverse order and for any Markov chain construct the equivalent electrical network?*

Exercise 4.25 *What is the probability of reaching vertex 1 before vertex 5 when starting a random walk at vertex 4 in each of the following graphs.*

1.



2.



Exercise 4.26 *Consider the electrical resistive network in Figure 4.14 consisting of vertices connected by resistors. Kirchoff's law states that the currents at each vertex 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.*

Exercise 4.27 *Consider the electrical network of Figure 4.15.*

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

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

3. *What is the effective resistance between a and b?*

4. *Convert the electrical network to a graph. What are the edge probabilities at each vertex so that the probability of a walk starting at c (d) reaches a before b equals the voltage at c (the voltage at d).?*

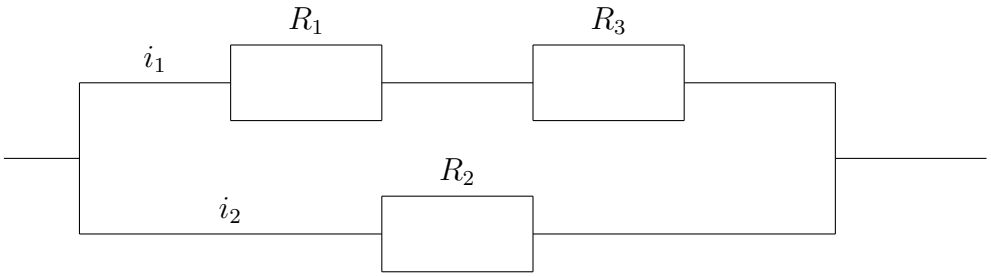


Figure 4.14: An electrical network of resistors.

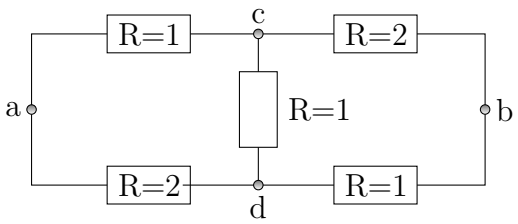


Figure 4.15: An electrical network of resistors.

5. *What is the probability of a walk starting at c reaching a before b? a walk starting at d reaching a before b?*

6. *What is the net frequency that a walk from a to b goes through the edge from c to d?*

7. *What is the probability that a random walk starting at a will return to a before reaching b?*

Exercise 4.28 *Consider a graph corresponding to an electrical network with vertices a and b. Prove directly that* $\frac{c_{eff}}{c_a}$ *must be less than or equal to one. We know that this is the escape probability and must be at most 1. But, for this exercise, do not use that fact.*

Exercise 4.29 *(Thomson's Principle) 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* $E = \frac{1}{2}\sum_{x,y} i_{xy}^2 r_{xy}$ *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 the distribution satisfying Ohm's law that minimizes energy dissipation.*

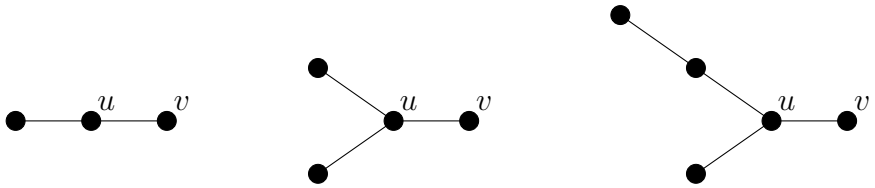


Figure 4.16: Three graphs

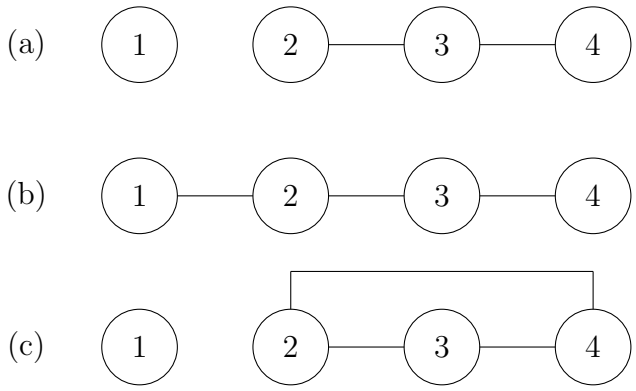


Figure 4.17: Three graph

Exercise 4.30 *(Rayleigh's law) 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. You may use Thomson's principle Exercise 4.29.*

Exercise 4.31 *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 4.32 *What is the hitting time h_{uv} for the three graphs if Figure 4.16.*

Exercise 4.33 *Show that adding an edge can either increase or decrease hitting time by calculating h_{24} for the three graphs in Figure 4.17.*

Exercise 4.34 *Consider the n vertex connected graph shown in Figure 4.18 consisting of an edge (u, v) plus a connected graph on $n - 1$ vertices and m edges. Prove that $h_{uv} = 2m + 1$ where m is the number of edges in the $n - 1$ vertex subgraph.*

Exercise 4.35 *Consider a random walk on a clique of size n. What is the expected number of steps before a given vertex is reached?*

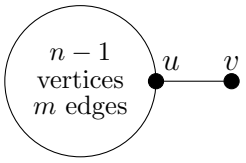


Figure 4.18: A connected graph consisting of $n-1$ vertices and m edges along with a single edge (u, v).

Exercise 4.36 *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 4.37 *Given the difference equation* $a_k t(i+k) + a_{k-1}t(i+k-1) + \cdots + a_1 t(i+1) + a_0 t(i) = 0$ *the polynomial* $a_k t^k + a_{k-i}t^{k-1} + \cdots + a_1 t + a_0 = 0$ *is called the characteristic polynomial.*

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

2. *If the roots of the characteristic polynomial are not distinct what is the most general form of the solution?*

3. *What is the dimension of the solution space?*

4. *If the difference equation is not homogeneous (i.e., the right hand side is not 0) and f(i) is a specific solution to the nonhomogeneous difference equation, what is the full set of solutions to the nonhomogeneous difference equation?*

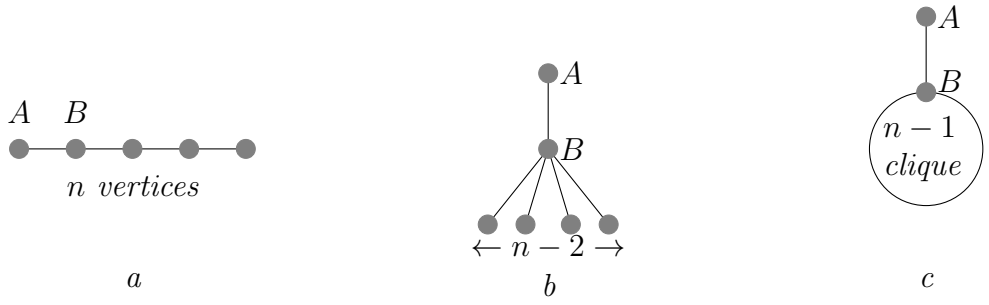
Exercise 4.38 *Show that adding an edge to a graph can either increase or decrease commute time.*

Exercise 4.39 *Consider the set of integers* $\{1, 2, \ldots, n\}$.

1. *What is the expected number of draws with replacement until the integer 1 is drawn.*

2. *What is the expected number of draws with replacement so that every integer is drawn?*

Exercise 4.40 *For each of the three graphs below what is the return time starting at vertex A? Express your answer as a function of the number of vertices, n, and then express it as a function of the number of edges m.*



Exercise 4.41 *Suppose that the clique in Exercise 4.40 was replaced by an arbitrary graph with $m-1$ edges. What would be the return time to A in terms of m, the total number of edges.*

Exercise 4.42 *Suppose that the clique in Exercise 4.40 was replaed by an arbitrary graph with $m-d$ edges and there were d edges from A to the graph. What would be the expected length of a random path starting at A and ending at A after returning to A exactly d times.*

Exercise 4.43 *Given an undirected graph with a component consisting of a single edge find two eigenvalues of the Laplacian $L = D - A$ where D is a diagonal matrix with vertex degrees on the diagonal and A is the adjacency matrix of the graph.*

Exercise 4.44 *A researcher was interested in determining the importance of various edges in an undirected graph. He computed the stationary probability for a random walk on the graph and let p_i be the probability of being at vertex i. If vertex i was of degree d_i, the frequency that edge (i, j) was traversed from i to j would be $\frac{1}{d_i}p_i$ and the frequency that the edge was traversed in the opposite direction would be $\frac{1}{d_j}p_j$. Thus, he assigned an importance of $\left|\frac{1}{d_i}p_i - \frac{1}{d_j}p_j\right|$ to the edge. What is wrong with his idea?*

Exercise 4.45 *Prove that two independent random walks starting at the origin on a two dimensional lattice will eventually meet with probability one.*

Exercise 4.46 *Suppose two individuals are flipping balanced coins and each is keeping tract of the number of heads minus the number of tails. At some time will both individual's counts be the same?*

Exercise 4.47 *Consider the lattice in 2-dimensions. In each square add the two diagonal edges. What is the escape probability for the resulting graph?*

Exercise 4.48 *Determine by simulation the escape probability for the 3-dimensional lattice.*

Exercise 4.49 *What is the escape probability for a random walk starting at the root of an infinite binary tree?*

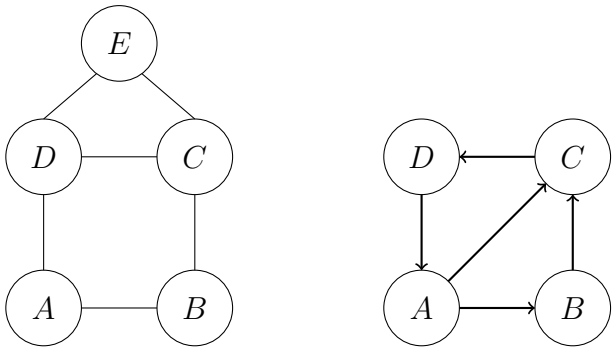


Figure 4.19: An undirected and a directed graph.

Exercise 4.50 *Consider a random walk on the positive half line, that is the integers* $0, 1, 2, \ldots$. *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 4.51 *Consider the graphs in Figure 4.19. Calculate the stationary distribution for a random walk on each graph and the flow through each edge. What condition holds on the flow through edges in the undirected graph? In the directed graph?*

Exercise 4.52 *Create a random directed graph with 200 vertices and roughly eight edges per vertex. Add* k *new vertices and calculate the pagerank with and without directed edges from the* k *added vertices to vertex 1. How much does adding the* k *edges change the pagerank of vertices for various values of* k *and restart frequency? How much does adding a loop at vertex 1 change the pagerank? To do the experiment carefully one needs to consider the pagerank of a vertex to which the star is attached. If it has low pagerank its page rank is likely to increase a lot.*

Exercise 4.53 *Repeat the experiment in Exercise 4.52 for hitting time.*

Exercise 4.54 *Search engines ignore self loops in calculating pagerank. Thus, to increase pagerank 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 4.55 *Number the vertices of a graph* $\{1, 2, \ldots, n\}$. *Define hitting time to be the expected time from vertex 1. In (2) assume that the vertices in the cycle are sequentially numbered.*

1. *What is the hitting time for a vertex in a complete directed graph with self loops?*
2. *What is the hitting time for a vertex in a directed cycle with* n *vertices?*

Create exercise relating strongly connected and full rank
Full rank implies strongly connected.
Strongly connected does not necessarily imply full rank

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Is graph aperiodic iff $\lambda_1 > \lambda_2$?

Exercise 4.56 *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 4.57 *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 as to how long or which files can be searched. Do not attempt an actual search without guidance from a knowledgeable person.*