Randomized Algorithms, Summer 2016 — Lecture 9 (11 pages)

Markov Chains: Graph Connectivity, Satisfiability, Rapid Mixing, Gambler's Ruin, Move-To-Front

Instructor: Thomas Kesselheim and Kurt Mehlhorn

1 Overview

We will apply our knowledge about random walks to the connectivity problem in graphs and to the satisfiability problem. We will see a connectivity algorithm that works in logarithmic space and a satisfiability algorithm that runs in expected time $\tilde{O}((4/3)^n)$ and hence is much faster than the naive 2^n algorithm that tries all possible assignments.

In the second part, we ask how fast a chain converges to its stationary distribution. Suppose we shuffle a deck of 52 cards by repeatedly selecting a random card and moving it to the top of the deck. The stationary distribution of this Markov chain is the uniform distribution. Let $\epsilon = 10^{-3}$ and let A be a subset of the state space. We start in state x. Let X_T be the state after T steps. How many steps do we have to make until

$$\mathbf{Pr}\left[X_T \in A \right] \leq |A| / 52! + \epsilon$$

for every A? Are 50 steps, 500 steps, 5000 steps sufficient?

To fill the two lectures, we consider two examples of Markov chains: gambler's ruin and the move-to-front heuristic for maintaining linear lists.

These lectures follow Sections 7.2 and 7.4.1 and Chapter 11 in Mutzenmacher/Upfal.

The material in these notes covers two lectures.

2 Connectivity

Given an undirected graph $G = (V, E)$ and two vertices s and t, we want to know whether there is a path from s to t. Depth-first search solves the problem in linear time and linear space. Alternatively, we may perform a random walk starting at s and report success when we reach t. More precisely, we perform a random walk of length $L \cdot 8nm$ for some parameter L.

Theorem 9.1. *If s and t are in the same connected component of G, the algorithm returns NO with probability at most 2^{-L}. The algorithm needs space $O(\log n + \log L)$.*

Proof. By the preceding lecture, the cover time of the graph is at most $4nm$, i.e., starting from any vertex we will reach t (in fact all vertices) in an expected number of at most $4nm$ steps. By Markov's inequality, we will reach t with probability $1/2$ within $8nm$ steps.

Divide time into epochs of length $8nm$. In each epoch, we reach t with probability at least $1/2$. Hence we do not reach t with probability at most $1/2$. Since there are L epochs, the probability that t is not reached is at most 2^{-L}.

The algorithm needs to keep track of the current vertex ($\log n$ bits) and the time ($O(\log n + \log L)$ bits). ☐

3 2-SAT

Let ϕ be a satisfiable 2-SAT formula, i.e., ϕ is a conjunction of clauses and each clause contains two literals. Let n be the number of variables. Consider the following algorithm.

Choose a random assignment A.
while A is not satisfying **do**

Let C be a clause that is not satisfied by A;
select one of the two variables in C at random and flip its value;
end while

Theorem 9.2. *If ϕ is satisfiable, the algorithm finds a satisfying assignment within an expected number of $4n(n+1)$ iterations.*

Proof. We observe first that the cover time of a chain graph of length $n+1$ (nodes 0 to n and edges $\{i, i+1\}$ for $0 \le i < n$) has cover time at most $4n(n+1)$.

Let S be a satisfying assignment, and let X be the Hamming distance between A and S, i.e., the number of variables for which A and S differ.

If $X = 0$, $A = S$, and we have found a satisfying assignment.

If $X = n$, flipping the value of any variable will reduce the distance by one.

Assume $1 \le X \le n-1$, and let C be the clause selected. Then A and S differ in the value of at least one of the variables in C; they may differ in both. Hence X is reduced by 1 with probability at least 1/2 and increased by 1 with probability at most 1/2.

Thus, we perform a walk on the chain graph where we have bias for going to a node with smaller index. Thus the walk will reach state 0 in an expected number of at most $4n(n+1)$ steps. $\square$

4 3-SAT

Let ϕ be a satisfiable 3-SAT formula, i.e., ϕ is a conjunction of clauses and each clause contains three literals. Let n be the number of variables. A first attempt is to generalize the algorithm of the preceding section in the obvious way.

Choose a random assignment A.
while A is not satisfying **do**
Let C be a clause that is not satisfied by A;
select one of the three variables in C at random and flip its value;
end while

We assume that ϕ is satisfiable and use S to denote a satisfying assignment. Let X_t be the number of variables in which A and S agree after t iterations. Then, for $1 \le j \le n-1$

$$\begin{aligned}
\mathbf{Pr}\left[\, X_{t+1} = j+1 \mid X_t = j \,\right] &\ge 1/3; \\
\mathbf{Pr}\left[\, X_{t+1} = j = 1 \mid X_t = j \,\right] &\le 2/3; \\
\mathbf{Pr}\left[\, X_{t+1} = n-1 \mid X_t = n \,\right] &= 1.
\end{aligned}$$

The distance shrinks with probability at least 1/3 since A and S must differ in at least variable appearing in clause C.

As before, we consider the process where we replace $\ge$ and $\le$ by equality.

$$\begin{aligned}
\mathbf{Pr}\left[\, Y_{t+1} = j+1 \mid Y_t = j \,\right] &= 1/3; \\
\mathbf{Pr}\left[\, Y_{t+1} = j-1 \mid Y_t = j \,\right] &= 2/3; \\
\mathbf{Pr}\left[\, Y_{t+1} = 1 \mid Y_t = 0 \,\right] &= 1.
\end{aligned}$$

Let h_j be the expected number of steps to reach n when starting from j. Then the following equations hold:

$$\begin{aligned}
h_n &= 0; \\
h_j &= 1 + \frac{2}{3}h_{j-1} + \frac{1}{3}h_{j+1}; \qquad \text{for } 1 \le j \le n-1 \\
h_0 &= 1 + h_1;
\end{aligned}$$

This is a system of $n+1$ linear equations in $n+1$ unknowns. It has a unique solution. Computing the solution for small values of n suggests that $h_j = h_{j+1} + f(j)$ for some function f. Clearly $f(0) = 1$. Also, for $1 \le j \le n-1$,

$$\begin{aligned} h_j &= 1 + \frac{2}{3}h_{j-1} + \frac{1}{3}h_{j+1} \\ &= 1 + \frac{2}{3}(h_j + f(j-1)) + \frac{1}{3}h_{j+1} && \text{induction hypothesis for } j-1, \end{aligned}$$

and hence

$$h_j = h_{j+1} + 3 + 2f(j-1).$$

Thus

$$f(j) = 3 + 2f(j-1) = 2^j f(0) + 3(1 + 2 + \ldots + 2^{j-1}) = 2^j + 3(2^j - 1) = 2^{j+2} - 3,$$

and hence

$$h_j = h_{j+1} + 2^{j+2} - 3 = 2^{n+2} - 2^{j+2} - 3(n-j).$$

Let us interpret this result. If we start with a good initial assignment (big j) the expected number of steps is small. However, if we start with an average initial assignment, say $j = n/2$, the expected number of steps is big. This suggests to try many initial assignments and for each one of them walk only a small number of steps. The following algorithm results.

for up to L times, terminating if all clauses are satisfied; **do**
  Choose a random assignment A;
  for up to $3n$ times, terminating when all clauses are satisfied; **do**
    Let C be a clause that is not satisfied by A;
    select one of the three variables in C at random and flip its value;
  end for
end for

Warning: I am changing the meaning of j. Before it was the number of variables in which A and S agree. Now it will be the number of variables in which they disagree.

Let q_j be the probability that the inner loop finds a satisfying assignment given that the Hamming distance between S and the initial assignment is j. This is certainly the case, if the inner loop finds the assignment within $3j$ steps. This is certainly the case, if among the first $3j$ steps, $2j$ are improving and j are deteriorating. Thus, for $j \ge 1$,

$$q_j \ge \binom{3j}{2j}\left(\frac{1}{3}\right)^{2j}\left(\frac{2}{3}\right)^{j}.$$

Also, $q_0 = 1$.

In order to estimate q_j we use $(n/e)^n \le n! \le (en)(n/e)^n$ for all n. Thus, for $j \ge 1$,

$$\begin{aligned} q_j &\ge \frac{(3j)!}{j!(2j)!}\left(\frac{1}{3}\right)^{2j}\left(\frac{2}{3}\right)^{j} \\ &\ge \frac{(3j/e)^{3j}}{(ej)(j/e)^j \cdot (e2j)(2j/e)^{2j}}\left(\frac{1}{3}\right)^{2j}\left(\frac{2}{3}\right)^{j} \\ &= \frac{3^{3j}2^j}{2e^2j^2 2^{2j}3^{2j}3^j} = \frac{1}{2e^2j^2 2^j} \ge \frac{1}{2e^2n^2 2^j}. \end{aligned}$$

This inequality is also true for $j = 0$. Let q be the probability that the inner loop finds a satisfying assignment when starting from a random assignment:

$$\begin{aligned} q &\geq \sum_{0\leq j\leq n} \mathbf{Pr}\left[\,\text{initial random assignment has distance } j \text{ from } S\,\right] \cdot q_j \\ &\geq \sum_{0\leq j\leq n} \binom{n}{j} 2^{-n} \frac{1}{2e^2 n^2 2^j} \\ &\geq \frac{1}{2e^2 n^2 2^n} \sum_{0\leq j\leq n} \binom{n}{j} \left(\frac{1}{2}\right)^j 1^{n-j} \\ &= \frac{1}{2e^2 n^2 2^n} \left(\frac{3}{2}\right)^n \\ &= \frac{1}{2e^2 n^2} \left(\frac{3}{4}\right)^n, \end{aligned}$$

where the next to last equality uses $\left(1 + \frac{1}{2}\right)^n = \sum_{0\leq j\leq n} \binom{n}{j} \left(\frac{1}{2}\right)^j 1^{n-j}$.

Theorem 9.3. *Assuming that ϕ is satisfiable, the algorithm finds a satisfying assignment after an expected number of $1/q = O(n^2(4/3)^n)$ iterations.*

The factor n^2 can be replaced by $\sqrt{n}$ by using a better approximation of the factorial function, namely,

$$\sqrt{2\pi m}(m/e)^m \leq m! \leq 2\sqrt{2\pi m}(m/e)^m.$$

5 Uniform Stationary Distribution for any Undirected Transition System

Let $G = (V, E)$ be an undirected connected graph. We will define a Markov chain whose stationary distribution is the uniform distribution. This is useful when we want to sample a state of the Markov chain (almost) uniformly. We simulate the chain for a sufficient number of steps. If the chain converges quickly to its stationary distribution, the state reached is close to being a random state.

Let N be the maximum degree of any vertex, and let M be any integer with $M \geq N$. For $x, y \in V$, let

$$P_{x,y} = \begin{cases} 1/M & \text{if } x \neq y \text{ and } (x,y) \in E; \\ 0 & \text{if } x \neq y \text{ and } (x,y) \notin E; \\ 1 - d(v)/M & \text{if } x = y. \end{cases}$$

Theorem 9.4. *If $M > N$, the chain is aperiodic. If the graph is connected, the chain is irreducible. If the chain is aperiodic and irreducible, the uniform distribution is the stationary distribution.*

Proof. If the graph is connected, one can go from any node to any other node. Thus the chain is irreducible. If $M > N$, the probability of staying in a node is nonzero. Hence the chain is aperiodic. Let π be the uniform distribution and let $\pi' = \pi^T P$. Then

$$\pi'_y = (\pi^T P)_y = \sum_{x;\ (x,y)\in E} \frac{1}{M}\pi_x + (1 - \frac{d(y)}{M})\pi_y = \frac{1}{n}\left(\frac{d(y)}{M} + 1 - \frac{d(y)}{M}\right) = \frac{1}{n}.$$

$\square$

6 Speed of Convergence to the Stationary Distribution

We know that an ergodic chain converges to its stationary distribution. How fast is the convergence process? Fast convergence is important for at least two reasons: it allows us to determine the stationary distribution by simulating the chain (this is how the page rank in search engines is computed), and it allows us to sample a state according to the stationary distribution by simulating the chain.

We first define a distance between distributions. We can then define the number of steps required to come ϵ-close to the stationary distribution. Next we introduce the concept of a coupling to prove rapid convergence.

6.1 Variation Distance

Let S be a set of states and let D_1 and D_2 be probability distributions on S. The *variation distance* between D_1 and D_2 is defined as

$$\|D_1 - D_2\| = \frac{1}{2} \sum_{s \in S} |D_1(s) - D_2(s)| .$$

Note that we simply add up the absolute values of all differences. The factor one-half guarantees that the variation distance is between 0 and 1. The following alternative characterization is easier to work with.

Lemma 9.5. *For a subset $S' \subseteq S$ and $i = 1, 2$, let $D_i(S') = \sum_{s \in S'} D_i(s)$. Then*

$$\|D_1 - D_2\| = \max_{S' \subseteq S} \left| D_1(S') - D_2(S') \right| .$$

Proof. Consider Figure 1.

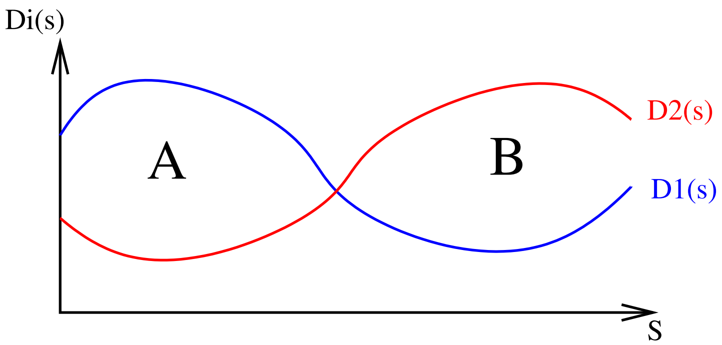


Figure 1: The figure illustrates two distributions over S. The height of the curve at s is equal to $D_i(s)$. The area under either curve is equal to one because we are dealing with distributions. Therefore the area A ($D_1(s) \geq D_2(s)$) is equal to the area B ($D_1(s) < D_2(s)$). The variation distance of D_1 and D_2 is equal to one half of the area of the union of A and B and hence equal to the area of A (or B). Thus $\|D_1 - D_2\| = |D_1(S^+) - D_2(S^+)| = |D_1(S^-) - D_2(S^-)| = \max_{S' \subseteq S} |D_1(S') - D_2(S')|$, where $S^+ = \{s \in S \mid D_1(s) \geq D_2(s)\}$ and $S^- = \{s \in S \mid D_1(s) < D_2(s)\}$.

□

Let π be the stationary distribution of a Markov chain with state space S and let p_x^t be the distribution on S obtained by running the chain for t steps steps starting in state x. Then $\Delta_x(t)$ denotes the variation distance between π and p_x^t and $\Delta(t)$ is the maximum of these values over all states x, i.e.,

$$\Delta_x(t) = \|p_x^t - \pi\|; \quad \Delta(t) = \max_{x \in S} \Delta_x(t).$$

We use $\tau_x(\epsilon)$ to denote the first step t such that $\Delta_x(t)$ is no more than ϵ and $\tau(\epsilon)$ the maximum of these values over all states x, i.e.,

$$\tau_x(\epsilon) = \min\{t \mid \Delta_x(t) \le \epsilon\}; \quad \tau(\epsilon) = \max_{x \in S} \tau_x(\epsilon).$$

The function τ is called the *mixing time of the chain.* If τ is polynomial in $\log(1/\epsilon)$ and the size of the problem, the chain is called *rapidly mixing.* As for the running time of algorithms, the size of a problem is a convention. For example, when shuffling cards, the size of the problem is the number of cards.

Lemma 9.6. *For an ergodic chain:* $\Delta(T+1) \le \Delta(T)$ *for all* T.

Proof. See Theorem 11.4 in Mutzenmacher/Upfal. □

6.2 Couplings

We next learn a powerful technique for bounding the mixing time of a chain. In a *coupling* of a Markov chain M_t with state space S, we run two instances of the chain in parallel. Formally, the coupling $Z_t = (X_t, Y_t)$ has state space $S \times S$ and satisfies:

$$\begin{aligned}\mathbf{Pr}\left[X_{t+1} = x' \mid Z_t = (x,y)\right] &= \mathbf{Pr}\left[M_{t+1} = x' \mid M_t = x\right];\\ \mathbf{Pr}\left[Y_{t+1} = y' \mid Z_t = (x,y)\right] &= \mathbf{Pr}\left[M_{t+1} = y' \mid M_t = y\right].\end{aligned}$$

Moreover, if $x = y$, both components make the same transition. When the two instances of the chain reach the same state, they are said to have *coupled.*

Note that each instance behaves like the original chain. However, the two instances are, in generally, dependent; see Figure **??**. In fact, coupling arguments are the art of finding the right dependence between the two instances.

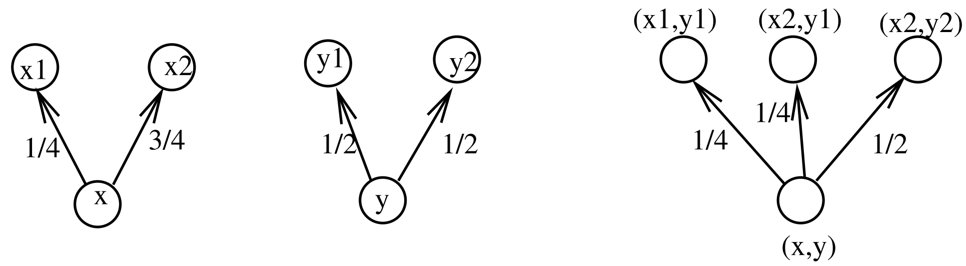


Figure 2: In M, we have the transitions out of states x and y shown on the left. The transitions in Z are shown on the right. Note that there is no transition from (x,y) to $(x1,y2)$, i.e., the two instances are not independent.

6.3 Shuffling Cards

Consider a deck of n cards. In each step, one of the cards is chosen uniformly at random and moved to the top of the deck. The state space has cardinality $n!$. The chain is ergodic. Prove it.

We use the following coupling: Let i be a random number between 1 and n. In the first instance, we select the i-th card from the top and move it to the top. Let C be this card. In the second instance, we move C to the top. Note that in both copies, the probability that a particular card is moved to the top is $1/n$, i.e., both copies are instances of the shuffling chain. If both instances are in the same state, they make the same move.

Lemma 9.7. *Assume that the instances start in states x and y, respectively. Let x_t and y_t be the states after t steps and let k_t be the number of distinct cards accessed in the first t steps. Then the top k_t cards of x_t and y_t are these k_t cards in the order of their last access (later accessed cards are nearer to the top).*

Proof. This is true before the first step since $k_0 = 0$. Consider step t. By induction hypothesis, $x_{t-1} = cx'_{t-1}$, $y_{t-1} = cy'_{t-1}$, where c consists of the k_{t-1} distinct cards accessed in the first $t-1$ steps. If the card accessed at time t was never accessed before, then the new common prefix is Cc, where C is the card accessed at time t. If the card was accessed before then $c = c'Cc''$ and the new common prefix is $Cc'c''$. □

We conclude that once all cards have been selected, the two instances are in the same state. By the coupon collector problem, if we run the chain for $n \ln n + cn$ steps, then the probability that a particular card was never chosen, is

$$\left(1 - \frac{1}{n}\right)^{n \ln n + cn} \leq e^{-(\ln n + c)} = \frac{e^{-c}}{n},$$

and hence the probability that some card was never chosen is no more than e^{-c}. Let $c = \ln(1/\epsilon)$. Then the probability that the chains have not coupled after $n \ln n + n \ln(1/\epsilon) = n \ln(n/\epsilon)$ steps is no more than ϵ. The following Lemma allows us to transfer this statement to the variation distance.

6.4 The Coupling Lemma

The variation distance between the distribution after T steps and the uniform distribution is bounded by the probability that the states after T steps are distinct. The coupling lemma captures this intuition.

Lemma 9.8 (Coupling Lemma)**.** *Let $Z_t = (X_t, Y_t)$ be a coupling for a Markov chain with state space S. For every integer T and positive real ϵ: if $\mathbf{Pr}\left[\, X_T \neq Y_T \mid X_0 = x,\ Y_0 = y \,\right] \leq \epsilon$ for all x and y in S then $\tau(\epsilon) \leq T$.*

Proof. Consider the coupling when X_0 is an arbitrary state x and Y_0 is chosen according to the stationary distribution π. Let $A \subseteq S$ be arbitrary. We show $\left|p_x^t(A) - \pi(A)\right| \leq \epsilon$. Then Lemma 9.5 implies that the variation distance between p_x^t and π is at most ϵ.

$$\begin{aligned}
p_x^T(A) = \mathbf{Pr}\left[\, X_T \in A \,\right] &\geq \mathbf{Pr}\left[\, (X_T = Y_T) \cap (Y_T \in A \,\right] \\
&= 1 - \mathbf{Pr}\left[\, (X_T \neq Y_T) \cup (Y_T \notin A) \,\right] \\
&\geq 1 - (\mathbf{Pr}\left[\, (X_T \neq Y_T) \,\right] + \mathbf{Pr}\left[\, (Y_T \notin A) \,\right]) \\
&= (1 - \mathbf{Pr}\left[\, (Y_T \notin A) \,\right]) - \mathbf{Pr}\left[\, (X_T \neq Y_T) \,\right] \\
&\geq \mathbf{Pr}\left[\, Y_T \in A \,\right] - \epsilon \\
&= \pi(A) - \epsilon,
\end{aligned}$$

where the second inequality uses the union bound and the last equality follows from the fact that y_T is distributed according to π; recall that y_0 is chosen according to the stationary distribution.

The same argument for the set $S - A$ shows $p_x^T(S \setminus A) \geq \pi(S \setminus A) - \epsilon$ and hence $p_x^T(A) \leq \pi(A) + \epsilon$. □

6.5 Shuffling Cards, Continued

Consider a deck of n cards. In each step, a random card is selected from the deck and moved to the top of the deck. After $T = n \ln n + n \ln(1/\epsilon) = n \ln(n/\epsilon)$ steps the variation distance between the distribution p_x^T and the uniform distribution is no more than ϵ. Here x is the initial state. In particular, for any subset A of states, $\left|p_x^T(A) - |A|/n!\right| \le \epsilon$. For $n = 52$ and $\epsilon = 10^{-3}$, the number of steps required is $52 \ln(52000) \le 580$.

6.6 Independent Sets

Let $G = (V, E)$ be an undirected graph and let Δ be the maximum degree of any vertex of G. We study a chain whose states are the independent sets of size k in G. *We will show that the chain is rapidly mixing provided that $k \le n/(3\Delta + 3)$.*

Let X_t be an independent set of size k. A move is made by choosing a random vertex $v \in X_t$ and a random vertex $w \in V$. Then

$$X_{t+1} = \begin{cases} (X_t \setminus v) \cup w & \text{if } (X_t \setminus v) \cup w \text{ is an independent set of size } k \\ X_t & \text{otherwise.} \end{cases}$$

We denote this transition as $move(v, w, X_t)$. Please verify that the chain is irreducible and ergodic and hence has a stationary distribution.

We consider the following coupling $Z_t = (X_t, Y_t)$. Let v be a random vertex in X_t and w a random vertex in V. In the first component we perform $move(v, w, X_t)$. For the second component, we determine a random vertex $v' \in Y_t$ as follows.

$$v' = \begin{cases} v & \text{if } v \in Y_t \cap X_t \\ \text{a random vertex in } Y_t \setminus X_t & \text{if } v \in X_t \setminus Y_t. \end{cases}$$

On Y_t, we perform $move(v', w, Y_t)$. Once both components are in the same state, they perform the same move. So we have a coupling.

Let $d_t = |X_t \setminus Y_t|$ be the number of elements in X_t that are not in Y_t. We show that d_t is more likely to decrease than increase in a step and use this to estimate the mixing time. Note that $\mathbf{Pr}[\, d_t > 0 \,]$ is the probability that the states at time t are distinct.

Assume $d_t > 0$. If $v \in X_t \cap Y_t$, and w is added to both or neither sets, the distance does not change. If w is added to exactly one of the sets, the distance may increase. If $v \in X_t \setminus Y_t$, the distance decreases if w is added to both sets, the distance does not change if w is added to neither set, and the distance does not increase if w is added to exactly one of the sets. We conclude:

- If $d_{t+1} = d_t + 1$ then $v \in X_t \cap Y_t$ ($k - d_t$ choices out of k choices) and w is chosen such that there is a transition in exactly one of the chains. Then w must be a vertex or a neighbor of a vertex in the set $(X_t \setminus Y_t) \cup (Y_t \setminus X_t)$. Thus

$$\mathbf{Pr}[\, d_{t+1} = d_t + 1 \mid d_t > 0 \,] \le \frac{k - d_t}{k} \cdot \frac{2d_t(\Delta + 1)}{n}.$$

- If $v \in X_t \setminus Y_t$ (d_t choices out of k choices) and w is chosen such that it neither a vertex or a neighbor of a vertex in the set $X_t \cup Y_t \setminus \{v, v'\}$ then $d_{t+1} = d_t - 1$. Note that $|X_t \cup Y_t| = k + d_t$. Thus

$$\mathbf{Pr}[\, d_{t+1} = d_t - 1 \mid d_t > 0 \,] \ge \frac{d_t}{k} \cdot \frac{n - (k + d_t - 2)(\Delta + 1)}{n}.$$

For $d_t > 0$, we thus have:

$$\begin{aligned}
\mathbf{Ex}\left[\, d_{t+1} \mid d_t \,\right] &= d_t + \mathbf{Pr}\left[\, d_{t+1} = d_t + 1 \mid d_t \,\right] - \mathbf{Pr}\left[\, d_{t+1} = d_t - 1 \mid d_t \,\right] \\
&\le d_t + \frac{k - d_t}{k} \cdot \frac{2d_t(\Delta + 1)}{n} - \frac{d_t}{k} \cdot \frac{n - (k + d_t - 2)(\Delta + 1)}{n} \\
&= d_t \cdot \left(1 - \frac{n - (3k - d_t - 2)(\Delta + 1)}{kn}\right) \\
&\le d_t \cdot \left(1 - \frac{n - (3k - 3)(\Delta + 1)}{kn}\right) \\
&\le d_t \cdot \left(1 - \frac{1}{kn}\right),
\end{aligned}$$

where the last inequality certainly holds for $k \le n/(3\Delta + 3)$. This inequality also holds for $d_t = 0$, since the two chains follow the same path once $d_t = 0$.

We obtain for $\mathbf{E}\left[\, d_{t+1} \,\right]$:

$$\begin{aligned}
\mathbf{E}\left[\, d_{t+1} \,\right] &= \sum_{d \ge 0} \mathbf{Ex}\left[\, d_{t+1} \mid d_t = d \,\right] \mathbf{Pr}\left[\, d_t = d \,\right] \\
&\le \sum_{d \ge 0} d \cdot \left(\frac{1}{kn}\right) \mathbf{Pr}\left[\, d_t = d \,\right] \\
&\le \mathbf{E}\left[\, d_t \,\right] \left(\frac{1}{kn}\right).
\end{aligned}$$

Induction then yields

$$\mathbf{E}\left[\, d_t \,\right] \le d_0 \left(\frac{1}{kn}\right)^t.$$

Since $d_0 \le k$ and d_t is a nonnegative integer, it follows that

$$\mathbf{Pr}\left[\, d_t \ge 1 \,\right] \le \mathbf{E}\left[\, d_t \,\right] \le k \left(\frac{1}{kn}\right)^{kn \cdot t/(kn)} \le ke^{-t/(kn)}.$$

and hence

$$\mathbf{Pr}\left[\, d_t \ge 1 \,\right] \le ke^{-t/(kn)} \overset{!}{\le} \epsilon,$$

provided that

$$t \ge kn \ln(k/\epsilon).$$

Thus

$$\tau(\epsilon) \le 1 + kn \ln(k\epsilon),$$

and $\tau(\epsilon)$ is polynomial in n and $\ln(1/\epsilon)$.

Theorem 9.9. *For $k \le n/(3\Delta + 3)$, the chain is rapidly mixing.*

7 Gambler's Ruin

We observe a gambler who repeatedly plays a fair game. In each round, he wins or looses a Euro with probability one-half. He stops playing when he either looses ℓ_1 Euros or wins ℓ_2 Euros.

The gambler gives rise to a Markov chain with states i, $-\ell_1 \le i \le \ell_2$. For $-\ell_1 < i < \ell_2$ we go to states $i - 1$ and $i + 1$ with probability $1/2$ each. States $-\ell_2$ and ℓ_1 are *absorbing*; the only transition is a self-loop and we take it with probability one. We start in state 0.

The states i, $-\ell_1 < i < \ell_2$ are *transient*, i.e., $\lim_{t\to\infty} P_i^t = 0$, where P_i^t is the probability of being in state i after t steps. With which probability do we end up in state ℓ_2.

Method 1: For each j, let q_j be the probability that we end up in state ℓ_2 when we start in state j. Then

$$\begin{aligned} q_{\ell_2} &= 1; \\ q_j &= (q_{j+1} + q_{j-1})/2 \qquad\qquad \text{for } -\ell_2 < j < \ell_1; \\ q_{-\ell_1} &= 0. \end{aligned}$$

This system is easy to solve. Rewrite the second equation as $q_{j+1} = 2q_j - q_{j-1}$ and apply it for $j = -\ell_1 + 1$ to ℓ_2. Then

$$\begin{aligned} q_{-\ell_1+2} &= 2q_{-\ell_1+1} - q_{-\ell_1} = 2q_{-\ell_1+1}; \\ q_{-\ell_1+3} &= 2q_{-\ell_1+2} - q_{-\ell_1+1} = 2q_{-\ell_1+2} - q_{-\ell+1} = 3q_{-\ell_1+1}; \\ q_{-\ell_1+4} &= 2q_{-\ell_1+3} - q_{-\ell_1+2} = 6q_{-\ell_1+2} - 2q_{-\ell+1} = 4q_{-\ell_1+1}; \end{aligned}$$

and hence by induction $1 = q_{\ell_2} = q_{-\ell_1+(\ell_1+\ell_2)} = (\ell_1+\ell_2)q_{\ell+1}$ or $q_{-\ell_1} = 1/(\ell_1+\ell_2)$ and thus $q_{-\ell_1+j} = j/(\ell_1+\ell_2)$. In particular $q_0 = \ell_1/(\ell_1+\ell_2)$.

Method 2: Let q be the probability of ending in state ℓ_2, and let W^t be the state after t steps (win of the player). Since the expected win of the player in each round is zero, $\mathbf{E}\left[\, W^t \,\right] = 0$ for all t. Also $\mathbf{E}\left[\, W^t \,\right] = \sum_{-\ell_1 \le i \le \ell_2} iP_i^t$ and hence

$$0 = \lim_{t\to\infty} \mathbf{E}\left[\, W^t \,\right] = \sum_{-\ell_1 \le i \le \ell_2} i \lim_{t\to\infty} P_i^t = -\ell_1(1-q) + \ell_2 q.$$

Thus,

$$q = \frac{\ell_1}{\ell_1+\ell_2}.$$

Exercise 1. *Redo the above for an unfair game. The player looses with probability* $2/3$ *and wins with probability* $1/3$.

8 Move-to-Front Heuristic for Maintaining Ordered Lists

Assume we store n items in a linear list. The i-th item is accessed with probability p_i and accessing an item in position j of the list has cost j. The probabilities are unknown to us. We may assume $p_1 \ge p_2 \ge \ldots \ge p_n$. If we know the probabilities we would store the items in order of decreasing probability and the expected access cost would be

$$Opt = \sum_{1 \le i \le n} ip_i.$$

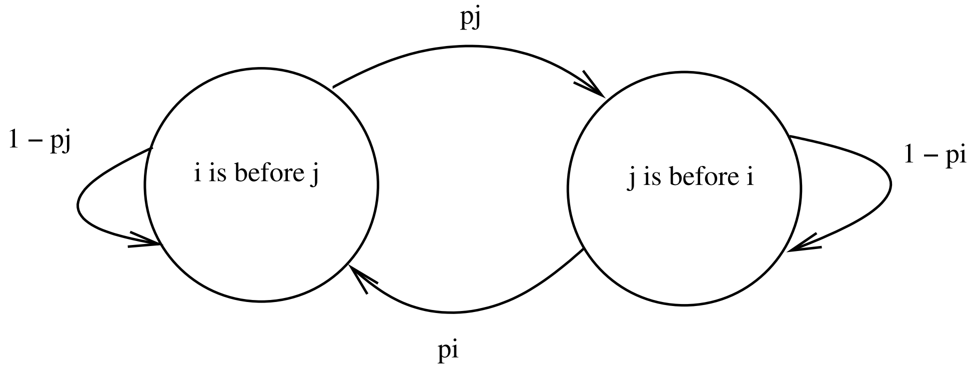
We use the *move-to-front* heuristic for maintaining the list. Whenever an item is accessed, we move it to the front of the list (all items preceding it before the access are moved back by one position). Our hope is that frequently accessed items tend to stay near the front of the list.

We can view the list as a Markov chain with $n!$ states; the states correspond to the $n!$ linear arrangements of n items. Let S be the set of states. The chain has a stationary distribution;

call it π. Assuming that the chain is in stationary distribution, the expected access cost is[1]

$$\begin{aligned} A &= \sum_{1\le i\le n} p_i \cdot \sum_{s\in S} \pi(s)\cdot \text{position of } i \text{ in } s \\ &= \sum_{1\le i\le n} p_i \cdot \sum_{s\in S} \left(\pi(s)\cdot\left(1+\sum_{j\ne i}[j \text{ is before } i \text{ in } s]\right)\right) \\ &= 1+\sum_{1\le i\le n} p_i \cdot \sum_{j\ne i} \sum_{\substack{s\in S\\ j \text{ is before } i \text{ in } s}} \pi(s) \end{aligned}$$

In order to compute the probability that j is before i, we split the set of states into two classes, one where i is before j and one where j is before i, and obtain a Markov chain with only two states. If an item different from i and j is accessed, we stay in the class, if item i is accessed, we move to the class (if not already there), where i precedes j, and if item j is accessed, we move to the class, where j precedes i.



It is easy to check that the stationary probability of having j before i is $p_j/(p_i+p_j$. Indeed,

$$\frac{p_j}{p_i+p_j} = p_j\frac{p_i}{p_i+p_j} + (1-p_i)\frac{p_j}{p_i+p_j}.$$

Plugging this expression into the expression for A yields

$$\begin{aligned} A &= 1+\sum_{1\le i\le n} p_i \cdot \sum_{j\ne i} \sum_{\substack{s\in S\\ j \text{ is before } i \text{ in } s}} \pi(s) \\ &= 1+\sum_{1\le i\le n} p_i \sum_{j\ne i} \frac{p_j}{p_i+p_j} \\ &= 1+\sum_{1\le i,j\le n,\ i\ne j} \frac{p_ip_j}{p_i+p_j} \\ &= 1+2\cdot\sum_{1\le i\le n} p_i \sum_{j<i} \frac{p_j}{p_i+p_j} \\ &\le 1+2\cdot\sum_{1\le i\le n} p_i(i-1) && \text{recall that } p_1\ge p_2\ge\ldots\ge p_n \\ &\le 2Opt. \end{aligned}$$

Remark 9.10. *Try to think of a similar scheme for trees. Accessed elements are always moved to the root. How will you rearrange the elements on the search path to the accessed element and the two children trees of the accessed element.*

Assume i precedes j in key-order. Does your scheme satisfy the following statement. i is an ancestor of j if since the last access to i, no key in $\{i+1,\ldots,j\}$ was accessed.

[1]The expression $[j$ is before i in $s]$ evaluates to 1 if j precedes i in s and evaluates to 0 otherwise.