6.856

Randomized Algorithms

2005/04/22

10 | 10 | 10/10

Dan Ports
`drkp@mit.edu`
Collaborators: {`sarahl`, `gpickard`}

Problem Set 10

Problem 1:

a) Let A be a $n \times n$ doubly-stochastic matrix. Then consider the uniform distribution $\pi = \frac{1}{n}\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}$. Then consider $\pi' = \pi A$. This is the sum of the rows of A divided by n. Since A is doubly-stochastic, the sum of each column is 1, so the sum of the rows is $\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}$, and so $\pi' = \frac{1}{n}\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix} = \pi$. So π is a stationary distribution.

b) Suppose $p_{ij} = p_{ji}$. Then A is a symmetric matrix by definition, so $A = A^T$. Each column of A sums to 1 since it is stochastic, so by symmetry each row also sums to 1, and it is doubly-stochastic. By the previous subproblem, the uniform distribution is stationary.

c) Now suppose $\pi_i p_{ij} = \pi_j p_{ji}$, and consider the individual terms of $\pi' = \pi A$. By definition, $\pi'_i = \sum_{j=1}^{n} \pi_j p_{ji}$. By the assumption, this is $\sum_{j=1}^{n} \pi_i p_{ij} = \pi_i \sum_{j=1}^{n} p_{ij}$. But since the matrix is stochastic, $\sum_{j=1}^{n} p_{ij} = 1$, and so $\pi' = \pi$. So π is a stationary distribution.

Problem 2:

We give an algorithm for coloring the vertices of a 3-colorable graph with two colors such that no triangle has all three vertices colored the same. We begin by coloring the vertices randomly with equal probability for each color. Then we identify any triangles that have all three vertices the same color. If any exist, we select a random one and change the color of one of its vertices, selected at random; note that this change may cause some other triangle to now have all vertices the same color. We repeat this process until no such conflicting triangle exists.

We analyze this process in the same way as the 2-SAT algorithm. Suppose the colors in the 2-coloring are c_1 and c_2. Let A be some 3-coloring of the graph, using colors c_1, c_2, and a third color c_3. Suppose that at some point during the execution of the algorithm, the assignment differs from A at i vertices. We select a random conflicting triangle that has all vertices the same color, and randomly change the color of one of them. Note that in the 3-coloring, each vertex has a different color. Assume *w.l.o.g.* that in the 2-coloring assignment each vertex has color c_1 (the argument is symmetric in the other case), so we change some vertex to c_2. If the randomly selected vertex is the one colored c_1 in A, the new assignment differs at $i+1$ vertices; if it is the one colored c_2, it now differs at $i-1$ vertices; and if it is the one colored c_3, it still differs at i vertices. All three possibilities occur with probability $1/3$.

So we can view this process as a Markov chain on a line graph of n vertices, with probability $1/3$ of increasing, decreasing, or staying at the same value, respectively. The process continues until no conflicting triangles remain. Certainly no conflicting triangles remain if the assignment is the same as A, i.e. $i = 0$; it can certainly be more, but this does not affect the runtime. The cover time for a line graph is $O(n^2)$, so $O(n^2)$ steps are required.

Problem 3:

*

We model the path of a car in New York as a random walk over a connected non-bipartite undirected graph $G = (V, E)$ (note that non-bipartiteness can be achieved by adding reflexive edges to the

OK

graph). We consider two simultaneous random walks over the same graph, starting from potentially different locations.

We can represent this as a random walk over a graph $\mathscr{G}$, where $\mathscr{G}$ has vertices given by the Cartesian product $V \times V$, and has an edge connecting vertices $\langle u,\ v\rangle$ and $\langle w,\ x\rangle$ if there exists both an edge from u to w and an edge from v to x in E. This models the two simultaneous random walks. Suppose the first random walk is at vertex u and the second is at vertex v; this is represented as $\langle u,\ v\rangle$ in $\mathscr{G}$. Then the first random walk moves to some neighbor w of u with probability $\frac{1}{d(u)}$ and similarly the second moves to some neighbor x of v with probability $\frac{1}{d(v)}$. The probability that both of these events take place is $\frac{1}{d(u)d(v)}$, which is also the probability of moving from $\langle u,\ v\rangle$ to $\langle w,\ x\rangle$ since $d(\langle u,\ v\rangle) = d(u)d(v)$ since $\mathscr{G}$ is the Cartesian product of G.

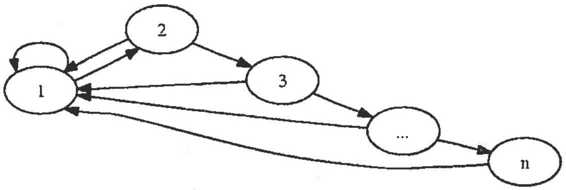
Note that $\mathscr{G}$ is also a non-bipartite graph: G is non-bipartite, so it contains some odd-length cycle. This cycle can be translated into a cycle in $\mathscr{G}$: simply start at the state corresponding to both walks being at some point in the cycle, then follow the transitions corresponding to both walks traveling along the odd cycle until they reach their starting locations.

Moreover, $\mathscr{G}$ is a connected graph. Consider any pair of points $\langle u,\ v\rangle$, $\langle w,\ x\rangle$ in $\mathscr{G}$. Since G is connected, there exist paths from u to w and from v to x in G. If both paths have the same length, we can translate them into a path in $\mathscr{G}$ in the obvious way. Otherwise, one of the paths is longer than the other, so we will need to lengthen the shorter path. If the difference in path lengths is even, we can simply repeatedly travel back and forth between two adjacent edges to increase the path lengths so they are even. Otherwise, if the difference in path lengths is odd, we take a detour through an odd-length cycle in the graph (which we proved existed); this makes the length difference even, reducing it to the previous case. We can do this for any pair $\langle u,\ v\rangle$, $\langle w,\ x\rangle$, so $\mathscr{G}$ is connected.

Now we have shown that two random walks on G can be modeled as a single random walk on a connected non-bipartite graph $\mathscr{G}$. Therefore, as shown in class, the expected cover time of $\mathscr{G}$ is $O\left(|V(\mathscr{G})|^3\right)$. Note that the number of vertices in $\mathscr{G}$ is the square of that in G. There exist states where both cars are in the same location, so the expected time until a collision is bounded by this time, which is $O\left(n^6\right)$.

Problem 4:

Consider the following directed graph, which consists of a chain of n vertices with directed edges between adjacent vertices, and directed edges from each vertex back to the first vertex in the chain:



This graph is clearly strongly connected, but its cover time is not polynomial in n. Suppose we start at vertex 1. To reach vertex n, we must choose the forward edges at each vertex, rather than the edge back to vertex 1. This occurs with probability $(1/2)^n$, since each choice happens with equal probability. Thus, the expected number of tries to reach vertex n, and so a bound on the expected cover time, is $O\left(2^n\right)$.

Problem 5:

Submitted separately.