Colorado CSCI 5454: Algorithms
Homework 8

Instructor: Bo Waggoner
Due: November 14, 2019 at 11:59pm
Turn in electronically via Gradescope.

Remember to list the people you worked with and any outside sources used (if none, write "none").

Problem 1 (6 points)

Given an unweighted, undirected graph G, let A_G be its adjacency matrix. Using $A_G^3 = A_G \cdot A_G \cdot A_G$, you will give an algorithm to count the number of distinct *triangles* in G.

(A triangle is a triple of vertices (a, b, c) such that (a, b), (b, c), and (c, a) are all edges of the graph.

Part a (4 points) Use A_G^3 to count the number of distinct triangles where (a, b, c) and (b, c, a) are different triangles (in other words, the order of vertices matters).

Part b (2 points) Now count the number of distinct triangles where (a, b, c), (b, c, a), (a, c, b), etc are all the same triangle.

Problem 2 (4 points)

Part a (4 points) You are given access to a list $A_G, A_G^2, A_G^3, \ldots, A_G^n$ of powers of the adjacency matrix an unweighted, undirected graph on n vertices. It takes constant time to query any entry $A_G^t(i, j)$.

Give an $O(\log(n))$ time algorithm to compute the length of the shortest path from vertex s to vertex t using this list.

Hint: first use the list to decide whether there exists any path at all from s to t.

Problem 3 (14 points)

(Programming assignment. Attach all code, printed to pdf.)

Part a (2 points) Come up with a nontrivial unweighted, undirected bipartite graph on at least 20 nodes. (No more than 100 is recommended.) This can be completely generated according to some rules or random process; or it can be inspired by some real-life situation or problem you're familiar with. Ensure the graph is connected.

Describe how you came up with the graph, and attach code used to generate it and/or read it into memory.

Part b (4 points) Let your bipartite graph be $G = (U, V, E)$. Execute 1000 random walks of length 20 with a biased starting distribution, as follows:

1. With probability $\frac{1}{4}$, start at a uniformly randomly chosen vertex from U. With the remaining probability, start at a uniformly randomly chosen vertex from V.

2. At each step, pick a neighbor of the current vertex uniformly at random and transition to that neighbor.

3. Repeat for 20 vertices.

Generate a plot where the horizontal axis is the step ($t = 1, 2, \ldots, 20$) and the vertical axis is the *fraction of the* 1000 *trials in which that step's vertex was in* U. For eaxmple, if there were 400 trials where the last vertex was in U and 600 trials where the last vertex was in V, then the plot would have value 0.4 above $t = 20$.

Briefly describe what you observe from this plot and why.

Part c (4 points) (*Lazy* random walk.) Repeat the previous experiment, but modify the random walk as follows. At each time step, with probability $\frac{1}{4}$, remain at the same vertex; otherwise, transition to a uniformly randomly chosen neighbor.

Generate the same plot for this experiment, briefly describe what you observe, whether it differs from the previous experiment, and if so, why.

Part d (4 points) (Limit of lazy walk.) Repeat the previous experiment, but now keep track of how many times the final vertex is vertex 1, 2, etc. Generate a histogram where the horizontal axis is the list of vertices and the vertical axis is the fraction of trials in which the final step of the walk was that vertex.

Then repeat this new experiment, but let each random walk have length only 10. Then repeat, but let each random walk have length 100.

Attach the three histograms. Describe your observations. How does the distribution of the final vertex change as we take more steps?

Part e (Bonus: 1 point) Compute the exact stationary distribution of the lazy random walk described above. Generate a histogram of this distribution in the same format as the three plots of the previous question, and briefly compare. Describe your approach to computing this stationary distribution and attach your code (you may use linear algebra software packages).

Problem 4 (6 points)

In the Metropolis-Hastings algorithm, we must construct an undirected graph on the state space. There were three important properties for the graph to satisfy. For each one, below, explain why it is important (or in other words, what goes wrong with the algorithm if it fails).

Part a (2 points) The maximum degree of any vertex, r, must be reasonably small.

Part b (2 points) If vertices u, v are neighbors in the graph, then $\frac{f(u)}{f(v)}$ should not be too small, and similarly for $\frac{f(v)}{f(u)}$.

Part c (2 points) The graph should have good *mixing time*: a random walk should converge quickly to the stationary distribution.

Problem 5 (10 points)

[Programming Metroplis-Hastings]

Consider a simplified system of n particles each of which is either in state 1 or state 0. Therefore, the state of the system can be described by a vector $x \in \{0,1\}^n$, i.e. a list of n bits.

Given a state x, let $p_x = \frac{\sum_i x_i}{n}$. In other words, p_x is the fraction of particles in state 1.

A physicist assures you that the probability $\pi(x)$ of any given state x is proportional to the *binary entropy* of p_x. The binary entropy function is

$$H_2(p) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}.$$

Part a (2 points) First, the physicist would like to sample configurations x from the distribution π described above. You will help her using Metropolis-Hastings.

First, describe an appropriate undirected, unweighted graph on the possible configurations of the system. How many states are there? Given a state x, how can one compute the list of its neighbors? What is the maximum degree r?

Part b (4 points) Implement the Metropolis-Hastings algorithm for the physicist. Your code should work for any n and any number of trials T. Attach the source code to the pdf.

Part c (4 points) For $n = 100$, run your sampling algorithm for at least $T = 10000$ trials.

To collect some statistics about the sampled states, store the fraction of ones, p_x, in each of the T trials Then plot a histogram of all the p_x variables.

Part d (Bonus: 1 point) Write down and plot the correct distribution over p_x when x is drawn from π. *Hint: how many states are there with k ones and $n-k$ zeros?* Compare to your Metropolis-Hastings histogram.