CS 229r - Spectral Graph Theory in Computer Science Prof. Salil Vadhan

Problem Set 2

Harvard SEAS - Fall 2020 *Due: Fri. Oct. 9, 2020 (5pm)*

Your problem set solutions must be typed (in e.g. LaTeX) and submitted on Gradescope. You are allowed 12 late days for the semester, of which at most 5 can be used on any individual problem set. (1 late day = 24 hours exactly). Please name your file `ps2-lastname.*`.

The problem sets may require a lot of thought, so be sure to start them early. You are encouraged to discuss the course material and the homework problems with each other in small groups (2-3 people). Identify your collaborators on your submission. Discussion of homework problems may include brainstorming and verbally walking through possible solutions, but should not include one person telling the others how to solve the problem. In addition, each person must write up their solutions independently, and these write-ups should not be checked against each other or passed around.

Strive for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details. Do not despair if you cannot solve all the problems! Difficult problems are included to stimulate your thinking and for your enjoyment, not to overwork you.

Problem 1.(brainstorm final project topics) Read the "Final Project Guidelines" document on the course website and submit about a paragraph as described in the "Topic Ideas" bullet.

Problem 2.(connectivity via random walks)

1. The *cover time* of a strongly connected digraph G is the maximum over all starting vertices a of the expected number of steps it takes for a random walk started at a to visit all the vertices of G. Prove that every simple undirected graph on n vertices has cover time at most $\mathrm{poly}(n)$. (Hint: use the fact that the mixing time of the random walk is $\mathrm{poly}(n)$ and be don't be afraid of paying extra polynomial factors; we are not asking you to compute a tight bound.)

2. Consider the problem of S-T Connectivity: given a digraph G, and two vertices s and t, decide whether there is a path from s to t in G. This can be solved in linear time using breadth-first or depth-first search. However, those algorithms also use a linear amount of space (i.e. memory). Using Part 1, design a *randomized* algorithm that solves S-T Connectivity on *undirected* graphs using only $O(\log n)$ bits of space. Your algorithm can freely toss coins, should halt on all sequences of coin tosses, and, on every input, should output the correct YES/NO answer with high probability over its coin tosses. We don't count the (read-only) input in the space complexity.

Problem 3.(understanding ω_π for directed graphs) Let G be a (possibly weighted) strongly connected digraph with random-walk transition matrix $W = MD_{out}^{-1}$ and stationary distribution π. Recall the definition:

$$\omega_\pi = \max_p \frac{\|Wp - \pi\|_\pi}{\|p - \pi\|_\pi},$$

where the maximum is taken over probability distributions p on the vertices of G and

$$\|x\|_\pi = \sqrt{\sum_{a \in V} \frac{x(a)^2}{\pi(a)}}.$$

1. Suppose that G is Eulerian, with $D_{out} = D = \text{diag}(\vec{d})$ and $\pi = \vec{d}/d(V)$. Prove that $\omega_\pi = \sqrt{\beta_2}$, where β_2 is as defined in Problem Set 1, Problem 3. (As you saw, the parameters $\beta_1, \ldots, \beta_n$ are the eigenvalues of the positive semidefinite matrix $A^T A$, and thus (by definition) $\sqrt{\beta_1}, \ldots, \sqrt{\beta_n}$ are the *singular values* of the normalized adjacency matrix A.)

2. Recall that $\tilde{\omega}_\pi$ is the analogue of ω_π for the lazy random walk $\tilde{W}$. Using Part 1 and the results of Problem Set 1, show that if G is a d-regular and unweighted graph on n vertices, then
$$\tilde{\omega}_\pi \leq 1 - \frac{1}{\text{poly}(n \cdot d)}.$$
(Contrast this with the fact that we can have $\omega_\pi = 1$ for 2-regular, unweighted, aperiodic G, as shown in section.) For extra credit, generalize to weighted and possibly irregular Eulerian digraphs, showing:
$$\tilde{\omega}_\pi \leq 1 - \frac{1}{\text{poly}(nd_{max}/w_{min})} = 1 - \frac{1}{\text{poly}(nw_{max}/w_{min})},$$
where d_{max} is the maximum (weighted) degree, and w_{max} and w_{min} are the maximum and minimum edge weights, respectively.

3. Now let G be a possibly non-Eulerian digraph. Consider a graph G' where we rescale the the weight on edge (a, b) to
$$w'(a, b) = w(a, b) \cdot \frac{\pi(a)}{d_{out}(a)} = W(b, a) \cdot \pi(a),$$
where $w(a, b)$ is the weight of edge (a, b) in G, $d_{out}(a)$ is the outdegree of vertex a in G, and $W(b, a)$ is the (b, a)'th entry of the random-walk matrix W. Show that G' is an Eulerian digraph whose random-walk matrix is also W. Why does this and Part 2 above not contradict the fact that lazy random walks on directed graphs can have exponential mixing time (as covered in section)?

Problem 4.(time-reversible Markov chains and MCMC) Let W be the transition matrix for an ergodic finite-state Markov chain (i.e. a random walk on a directed, strongly connected, and aperiodic graph). A probability distribution π is said to satisfy the *detailed balance equations* if for all a, b,

$$W(b, a) \cdot \pi(a) = W(a, b) \cdot \pi(b).$$

1. Prove that if π satisfies the detailed balance equations, then π is in fact the (unique) stationary distribution for G. (An ergodic Markov chain whose stationary distribution satisfies the detailed balance equations is called *time-reversible.*)

2. Show that a Markov chain with transition matrix W is time-reversible if and only if there exists an undirected graph whose random-walk matrix is W.

3. In class, we defined a time-reversible Markov chain on the set $\mathcal{F}$ of spanning forests of a simple, undirected graph G such that uniform distribution on $\mathcal{F}$ is stationary. Modify the transition probabilities to obtain a time-reversible Markov chain for the case that G is weighted and we want the stationary distribution to sample a forest $F \in \mathcal{F}$ with probability proportional to its weight $w(F) = \prod_{e \in F} w_e$. (Hint: you may find it useful to include appropriately weighted self-loops.)

Problem 5.(extra credit: mixing vs. 2nd eigenvalue) Let G be an undirected, possibly weighted graph whose random-walk transition matrix $W = MD^{-1}$ has eigenvalues $1 = \omega_1 \geq \omega_2 \geq \cdots \geq \omega_n$. We have seen that from any start vertex, the random walk on G will get to within ℓ_1 distance ε of stationary in time

$$t = O\left(\frac{\log(nd_{max}/\varepsilon d_{min})}{1 - \omega_\pi}\right),$$

where $\omega_\pi = \max\{\omega_2, -\omega_n\}$, which equals ω_2 in the case of a lazy random walk.

Here you'll prove a converse bound (except for the $\log(nd_{max}/d_{min})$ term in the numerator). Specifically, show that G has a vertex a such that the random walk from a takes at least

$$t = \Omega\left(\frac{\log(1/\varepsilon)}{1 - \omega_2}\right)$$

steps to get to within ℓ_1 distance ε of stationary, assuming $\omega_2 \geq 1/2$.

Hint: x be a *left* eigenvector of W of eigenvalue ω_2 and let a be a vertex maximizing $|x(a)|$. Use the fact that $D^{1/2}x$ is orthogonal to $\overrightarrow{d^{1/2}}$.