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

Problem Set 3

Harvard SEAS - Fall 2020 — *Due: Fri. Oct. 23, 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 `ps3-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.(Final Project Next Step) You will have received comments on your initial project ideas from problem set 2. Using these comments, and rereading the "Final Project Guidelines" document on the course website (https://salil.seas.harvard.edu/classes/spectral-graph-theory), refine your topic ideas and, if you wish, seek out 1-2 collaborators on Piazza. Your group should submit a project proposal (or two) approximately a half-page long. (All group members should submit the same proposal.) Include at least three citations to relevant background reading or related research literature, and be sure to indicate the nature of the project (e.g. synthesizing material that we will not cover in class, implementing and running experiments with algorithms using spectral graph theory, trying to obtain some small new theoretical results, etc.) Try to anticipate any challenges you might encounter, and ask concrete questions where you could use additional pointers or guidance from us.

Problem 2.(expander decompositions) In this problem, you'll see how to use Fiedler's Algorithm to partition an arbitrary graph into reasonably good expanders (as measured by ν_2, the second smallest eigenvalue of the normalized Laplacian), while cutting few edges. Efficient implementations of expander decompositions have found many algorithmic applications in recent years; intuitively they allow us to reduce solving a problem on an arbitrary graph to solving it on expanders, which is often an easier task.

Specifically, consider the following algorithm on input an unweighted, undirected graph $G = (V, E)$ and a parameter $\nu > 0$:

1. Initialize $E' = E$ and $G' = (V, E')$.
2. While $G' = (V, E')$ contains a connected component $G_C = (C, E_C)$ such that $\nu_2(G_C) < \nu$, do:
   (a) Use Fiedler's Algorithm to find a cut $(S, C - S)$ of small conductance in G_C.

(b) Remove the edges between S and $C - S$ from E'.

Observe that the above algorithm can be implemented in polynomial time, and outputs a final graph $G' = (V, E')$ in which every connected component G_C has $\nu_2(G_C) \geq \nu$.

Show that for any sufficiently small $\varepsilon > 0$ and an appropriate setting of $\nu = \Theta(\varepsilon/\log m)^2$, the above algorithm will guarantee that $|E'| \geq (1 - \varepsilon) \cdot m$. In particular, we can cut only an $\varepsilon = 1/\text{polylog}(n)$ fraction of the edges and ensure that on every component, the lazy random walk mixes in time at most $\text{polylog}(n)$. (Hint: in the analysis, start the algorithm by assigning each edge a charge of 1, and whenever removing edges in a cut $(S, C - S)$, redistribute the charges of the cut edges uniformly over the edges of S or $C - S$, whichever has fewer edges, to maintain a total charge of m. Argue that each time the charge of an edge is increased, it increases by a multiplicative factor of at most $1 + O(\sqrt{\nu})$.)

Problem 3.(pseudoinverses) Let A be an $n \times n$ complex matrix, with a singular value decomposition $A = U\Sigma V^*$ for unitary matrices U and V, a diagonal, nonnegative real matrix Σ. The *pseudoinverse* of A is the matrix $A^+ = V\Sigma^+U^*$, where Σ^+ is the diagonal matrix obtained by replacing each diagonal entry σ of Σ with

$$\sigma^+ = \begin{cases} 0 & \text{if } \sigma = 0 \\ 1/\sigma & \text{otherwise} \end{cases}.$$

(It can be shown that A^+ does not depend on the choice of singular-value decomposition of A.)

1. Prove that if the linear system $Ax = b$ has a solution, then $x = A^+b$ is a solution.

2. Show that if A is a Hermitian matrix, then A^+ is the unique matrix B with the following property: for every eigenvector v of A of eigenvalue λ, v is an eigenvector of B of eigenvalue λ^+.

3. Let G be a connected, undirected graph with normalized Laplacian N. We saw when discussing the power method that it is useful to have fast algorithms for applying N^+ to a vector. Show that it suffices to be able to do this for regular graphs. Specifically, let G_{reg} be obtained by adding appropriately weighted self-loops to G to make it regular and let N_{reg} be its normalized Laplacian. Show that there are diagonal matrices D_1 and D_2 such that $N^+ = D_1 N_{reg}^+ D_2$. (Hint: relate N^+ to L^+ and observe the effect of adding self-loops on L.)

4. Let G be a connected, undirected, regular graph with normalized Laplacian $N = I - W$. Prove that
$$N^+ = (I - W)^+ = \frac{1}{2}\left(I - J + (I + W)(I - W^2)^+(I + W)\right),$$
where J is the matrix in which every entry is $1/n$. Thus computing or applying the pseudoinverse of $I - W$ reduces to applying the pseudoinverse of $I - W^2$. This recursion will be the basis of a nearly linear time algorithm that we see later in the course.

Problem 4.(Cayley expanders)

1. For a prime p and $k \in \mathbb{N}$, consider the group $\mathbb{Z}_p^k$, whose elements are k-dimensional vectors $a = (a_0, \ldots, a_{k-1})$ where each $a_i \in \{0, 1, \ldots, p-1\}$ and the arithmetic is coordinate-wise addition modulo p:

$$(a_0, \ldots, a_{k-1}) + (b_0, \ldots, b_{k-1}) = (a_0 + b_0 \bmod p, \ldots, a_{k-1} + b_{k-1} \bmod p).$$

   Consider the following set $S \subseteq \mathbb{Z}_p^k$:

$$S = \{(\alpha, \alpha\beta, \alpha\beta^2, \ldots, \alpha\beta^{k-1}) : \alpha, \beta \in \mathbb{Z}_p\},$$

   where again all arithmetic is modulo p.

   Prove that the graph $G = \mathrm{Cay}(\mathbb{Z}_p^k, S)$ has $\omega(G) \leq (k-1)/p$.

   (Hint: you may use the facts that (a) a nonzero polynomial over $\mathbb{Z}_p$ of degree at most $k-1$ has at most $k-1$ roots in $\mathbb{Z}_p$[1] and (b) if γ is a nonzero element of $\mathbb{Z}_p$, then $\{\alpha\gamma : \alpha \in \mathbb{Z}_p\} = \mathbb{Z}_p$.)

   Conclude that for infinitely many n there is an explicit graph on n vertices with spectral expansion at least $1/2$ and degree at most $O(\log^2 n)$.[2]

2. (extra credit) Let Γ be an abelian group of size n, and let $G = \mathrm{Cay}(\Gamma, S)$ for $|S| = d$. Prove that the diameter of G is at least $n^{1/d} - 1$ (or something similar, you do not need to get this exact expression). (Hint: use commutativity!) Conclude that if G has spectral expansion γ for a constant $\gamma > 0$, then

$$d = \Omega(\log n / \log\log n).$$

   Thus abelian groups cannot yield Cayley expanders of constant degree.

[1] That is, for $f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{k-1} x^{k-1}$ where each $c_i \in \mathbb{Z}_p$ and not all c_i's are zero, there are at most $k-1$ values $\beta \in \mathbb{Z}_p$ such that $f(\beta) \bmod p = 0$).

[2] Those familiar with algebraic error-correcting codes may note that this construction works over any field, not just prime fields, and when we use a field of characteristic 2, the Cayley expander we obtain amounts to applying the correspondence between linear codes and Cayley graphs from Problem Set 1 to the code obtained by concatenating a Reed–Solomon code and a Hadamard code.