CSE 521: Design and Analysis of Algorithms **Fall 2021**

Problem Set 4

Deadline: Dec 9th in gradescope

1) Let G be a connected (unweighted undirected) graph with n vertices and conductance $\phi(G)$. In this exercise we see how to approximate the diameter of G, $\text{diam}(G) := \max_{u,v} d(u,v)$ where $d(u,v)$ is the shortest path from u to v, using eigenvalues of the normalized Laplacian matrix.

   - Show that
   $$\text{diam}(G) \le O(\log n/\phi(G))$$
   Hint: For a set $S \subseteq V$ let $N(S) = \{v : \exists u \in S, (u,v) \in E\}$ be the set of neighbors of S. Show that
   $$\text{vol}(N(S) \cup S) \ge (1 + \phi(S)) \,\text{vol}(S)$$
   - Prove that $diam(G) \le O(\log n/\lambda_2(\tilde{L}))$ where $\tilde{L}$ is the normalized Laplacian matrix and $\lambda_2(\tilde{L})$ is its second smallest eigenvalue.

2) Given a (undirected connected) graph G, with Laplacian matrix, the effective resistance between a pair of vertices i,j of G is defined as $\text{Reff}(i,j) = b_{i,j}^T L^\dagger b_{i,j}$ where $b_{i,j} = 1_i - 1_j$ is the vector that is $+1$ at i-th coordinate, -1 at j-th coordinate and zero everywhere else. Note that since the Laplacian matrix has a zero eigenvalue its inverse is not well-defined; that is why we define the effective resistance with respect to a pseudo-inverse of the Laplacian. Namely if $L = \sum_i \lambda_i v_i v_i^T$, then $L^\dagger := \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^T$.

   So, in particular, the effective resistance of an edge $e = (i,j)$ is $\text{Reff}(e) := \text{Reff}(i,j)$. We observed that the effective resistance can be exploited to generate a spectral sparsifier of a graph. In this exercise we prove that effective resistance defines a metric over vertices of G. This fact has numerous applications in algorithmic use of effective resistance.

   (a) Prove that the $\text{Reff}(i,j) \ge 0$ and that $\text{Reff}(i,j) = \text{Reff}(j,i)$ for all i,j.

   (b) Suppose that $b_{i,j} = Lp$ where L is the Laplacian and $p \in R^n$. Show for any $k \neq i,j$, $p_i \ge p_k \ge p_j$, i.e., p_i has largest value in p and p_j has smallest value in p.

   (c) Use the previous part to show that for any three distinct vertices i,j,k
   $$b_{i,j}^T L^\dagger b_{j,k} \le 0.$$

   (d) Use the previous part to prove that effective resistance defines a metric: namely for any three distinct vertices i,j,k,
   $$\text{Reff}(i,j) + \text{Reff}(j,k) \ge \text{Reff}(i,k).$$

3) In this problem we use expander graphs to design a much faster algorithm for problem 1 of HW2. We will use the following theorem:

   Theorem 4.1. *Let G be a d-regular expander graph where λ is the second smallest eigenvalue of the normalized Laplacian matrix. Consider the following random walk: Let X_1 be a uniformly random vertex of G, each time given X_i, X_{i+1} will be a uniformly random neighbor of X_i, i.e., starting from X_1 we run a simple random on G. Suppose we run this walk for t time steps. Then, for any set $S \subseteq V$,*
   $$\mathbb{P}\left[\left|\frac{1}{t}|\{X_1, \dots, X_t\} \cap S| - |S|/n\right| > \epsilon\right] \le 2\exp(-\lambda t \epsilon^2).$$

Say A is a randomized algorithm that uses m random bits and will output the optimum of a minimization problem with probability at least 1/2. Let G be a d-regular expander graph (for a constant d) with $n = 2^m$ many vertices and $\lambda > 1/10$[1].

(a) Prove that to run a walk of length t on G, as described above, we only need $m + O(t \log d)$ many random bits (assuming d is an absolute constant).

(b) Prove that for any integer $r < m$ we can improve the success probability of A to $1 - 1/2^r$ by running A only $O(r)$ many times using only $O(m)$ many bits. In this part treat d as a constant and ignore any factor of d in the $O(.)$ notation.

So, we need significantly less random bits compared to the simplest approach which uses $O(rm)$ many random bits.

4) Implement the spectral sparsification algorithm that we discussed in class. Print your pseudo-code in the pdf-file.

5) You are given data containing grades in different courses for 5 students; say $G_{i,j}$ is the grade of student i in course j. (Of course, $G_{i,j}$ is not defined for all i, j since each student has only taken a few courses.) We are trying to "explain" the grades as a linear function of the student's innate aptitude, the easiness of the course and some error term.

$$G_{i,j} = \text{aptitude}_i + \text{easiness}_j + \epsilon_{i,j},$$

where $\epsilon_{i,j}$ is an error term of the linear model. We want to find the best model that minimizes the sum of the $|\epsilon_{i,j}|$'s.

a) Write a linear program to find aptitude$_i$ and easiness$_j$ for all i, j minimizing $\sum_{i,j} |\epsilon_{i,j}|$.

b) Use any standard package for linear programming (Matlab/CVX, Freemat, Sci-Python, Excel etc.; we recommend CVX on matlab) to fit the best model to this data. Include a printout of your code, the objective value of the optimum, $\sum_{i,j} |\epsilon_{i,j}|$, and the calculated easiness values of all the courses and the aptitudes of all the students.

	MAT	CHE	ANT	REL	POL	ECO	COS
Alex		C-	B	B+	A	C+	
Billy	B-	A-	C		A+	D+	B
Chris	B+		B+	C		C	B+
David	A+		B-	A-		A-	
Elise		B-	D+	A-		D	D

Assume $A = 4, B = 3$ and so on. Also, let $B+ = 3.33$ and $A- = 3.66$.

6) **Extra Credit.** In this problem we see applications of expander graphs in coding theory. Error correcting codes are used in all digital transmission and data storage schemes. Suppose we want to transfer m bits over a noisy channel. The noise may flip some of the bits; so 0101 may become 1101. Since the transmitter wants that the receiver correctly receives the message, he needs to send $n > m$ bits encoded such that the receiver can recover the message even in the presence of noise. For example, a naive way is to send every bit 3 times; so, 0101 becomes 000111000111. If only 1 bit were flipped in the transmission receiver can recover the message but even if 2 bits are flipped, e.g., 110111000111 the recover is impossible. This is a very inefficient coding scheme.

[1] Here is a simple construction of a 3-regular expander graph: Take p be a prime, let $V = \{0, 1, \ldots, p-1\}$. Each vertex i is connected to $i - 1 \mod p$, $i + 1 \mod p$, and to its multiplicative inverse, a^{-1}, mod p. It is known that the second smallest eigenvalue of normalized Laplacian such a graph is at least ϵ for some explicit ϵ

An error correcting code is a mapping $C : \{0,1\}^m \to \{0,1\}^n$. Every string in the image of C is called a codeword. We say a coding scheme is linear, if there is a matrix $M \in \{0,1\}^{(n-m)\times n}$ such that for any $y \in \{0,1\}^n$, y is a codeword if and only if

$$My = 0.$$

Note that we are doing addition and multiplication in the field F_2.

a) Suppose C is a linear code. Construct a matrix $A \in \{0,1\}^{n\times m}$ such that for any $x \in \{0,1\}^m$, Ax is a code word and that for any distinct $x, y \in \{0,1\}^m$, $Ax \neq Ay$.

The rate of a code C is defined as $r = m/n$. Codes of higher rate are more efficient; here we will be interested in designing codes with r being an absolute constant bounded away from 0. The Hamming distance between two codewords c^1, c^2 is the number of bits that they differ, $\|c^1 - c^2\|_1$. The minimum distance of a code is $\min_{c^1,c^2} \|c^1 - c^2\|_1$.

b) Show that the minimum distance of a linear code is the minimum Hamming weight of its codewords, i.e., $\min_c \|c\|_1$.

Note that if C has distance d, then it is possible to decode a message if less than $d/2$ of the bits are flipped. The minimum relative distance of C is $\delta = \frac{1}{n} \min \|c^1 - c^2\|_1$. So, ideally, we would like to have codes with constant minimum relative distance; in other words, we would like to say even if a constant fraction of the bits are flipped still one can recover the original message.

Next, we describe an error correcting code scheme based on bipartite expander graphs with constant rate and constant minimum relative distance. A $(n_L, n_R, D, \gamma, \alpha)$ expander is a bipartite graph $G(L \cup R, E)$ such that $|L| = n_L, |R| = n_R$ and every vertex of L has degree D such that for any set $S \subseteq L$ of size $|S| \leq \gamma n_L$,

$$N(S) \geq \alpha |S|.$$

In the above, $N(S) \subseteq R$ is the number of neighbors of vertices of S. One can generate the above family of bipartite expanders using ideas similar to Problem 1. We use the following theorem without proving it.

Theorem 4.2. *For any $\epsilon > 0$ and $m \leq n$ there exists $\gamma > 0$ and $D \geq 1$ such that a $(n, m, D, \gamma, D(1-\epsilon))$-expander exists. Additionally, $D = \Theta(\log(n_L/n_R)/\epsilon)$ and $\gamma n_L = \Theta(\epsilon n_R/D)$.*

Now, we describe how to construct the matrix M. We start with a $(n_L, n_R, D, \gamma, D(1-\epsilon))$ expander for $n_L = n, n_R = n - m$. For our calculations it is enough to let $n = 2m$. We name the vertices of L, $\{1, 2, \ldots, n\}$; so each bit of a codeword corresponds to a vertex in L. We let $M \in \{0,1\}^{(n-m)\times n}$ be the Tutte matrix corresponding to this graph, i.e., $M_{i,j} = 1$ if and only if the i-th vertex in R is connected to the j-th vertex in L. Observe that by construction this code has rate $1/2$. Next, we see that δ is bounded away from 0.

c) For a set $S \subseteq L$, let $U(S)$ be the set of unique neighbors of S, i.e., each vertex in $U(S)$ is connected to exactly one vertex of S. Show that for any $S \subseteq L$ such that $|S| \leq \gamma n$,

$$|U(S)| \geq D(1 - 2\epsilon)|S|.$$

d) Show that if $\epsilon < 1/2$ the minimum relative distance of C is at least γ.

The decoding algorithm is simple to describe but we will not describe it here.