CSE 521: Design and Analysis of Algorithms **Fall 2020**

Problem Set 4

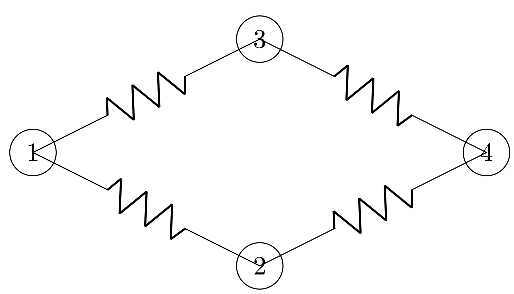
Deadline: Dec 8th in Canvas

1) Let G be a graph with maximum degree $\Delta := \max_v d(v)$ for any $v \in V$. Show that

$$\lambda_{\max}(A) \geq \sqrt{\Delta}.$$

Hint: Recall that by Rayleigh quotient, $\lambda_{\max} = \max_x \frac{x^T Ax}{x^T x}$. So, to lower bound $\lambda_{\max}$ it is enough to construct a vector x such that $x^T Ax \geq \sqrt{\Delta} x^T x$.

2) Given a connected graph $G = (V, E)$ you can construct an electrical network by replacing every edge with an effective resistance of resistance 1. Given any two vertices $u, v \in V$, the *effective resistance* between u, v is the resistance that you can put between u, v such that the energy of any electrical flow sent from u to v is the same as the original network. For example, the effective resistance between $2, 3$ in the following network is 1. It turns out that for any pair of vertices $u, v \in V$, the effective resistance between u, v is



equal to $b_{u,v}^T L^\dagger b_{u,v}$ where $b_{u,v} = \mathbf{1}_u - \mathbf{1}_v$ and $L^\dagger$ is the pseudo inverse of L; if $L = \sum_{i=1}^n \lambda_i v_i v_i^T$, then $L^\dagger = \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^T$. Note that since L has a zero eigenvalue, its inverse is not well-defined.

It is a well-known fact that in any **connected** graph with n vertices the sum of the effective resistances of all edges is equal to $n - 1$. Here, we prove this fact: Show that

$$\sum_{(u,v) \in E} b_{u,v}^T L^+ b_{u,v} = n - 1.$$

3) Given a graph $G = (V, E)$, recall that $M \subseteq E$ is a matching if no two edges in M have the same endpoint. We say a matching M is *maximal* if there is no matching $M' \supsetneq M$, i.e., there is no way to add an edge to M and get a larger matching. Note that a maximal matching is not necessarily a maximum matching.

   a) Show that for any n, the cycle C_{2n} has a maximal matching M of size $|M| \leq 2n/3 + 1$. Note that the size of a *maximum* matching of C_{2n} is n.

   b) Let G be a d-regular graph with $d = \lambda_1 \geq \cdots \geq \lambda_n \geq -d$ be eigenvalues of the adjacency matrix of G. Let $\lambda^* = \max\{\lambda_2, |\lambda_n|\}$. A d-regular strong expander graph satisfies $\lambda^* \leq O(\sqrt{d})$. It is known that a random d-regular graph is a strong expander with high probability.

   Use the expander mixing lemma to show that any maximal matching of G has size at least $(1/2 - O(\lambda^*/d))n$. In other words, it is very easy to approximate the maximum matching problem in random graphs, any maximal matching would work.

Theorem 4.1 (Expander Mixing Lemma)**.** *Let G be a d-regular graph and $d = \lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq -d$ be the eigenvalues of the adjacency matrix of G, A. Let $\lambda^* = \max\{\lambda_2, |\lambda_n|\}$. Then, for any two (non necessarily disjoint) sets $S, T \subseteq V$,*

$$\left| |E(S,T)| - \frac{d \cdot |S| \cdot |T|}{n} \right| \leq \lambda^* \sqrt{|S||T|}.$$

Here $E(S,T) = \{(u,v) \in E : u \in S, v \in T\}$.

Note that $d|S||T|/n$ is the expected number of edges between S, T in a random graph where there is an edge between each pair of vertices u, v with probability d/n. So, the above lemma says that in a strong expander graph, for any large enough sets $|S|, |T|$, the number of edges between S, T is very close to what you see in a random graph in expectation.

4) In this problem you are supposed to implement the spectral partitioning algorithm that we discussed in class. You are given a giant network, "com-Amazon" input in https://snap.stanford.edu/data/ and you should find a sparse cut in this network. My code has found a cut of sparsity about 0.7%. Note that since the graph is huge you need to carefully store the edges of this graph. You should also use the power method to find the 2nd smallest eigenvalue of the normalized Laplacian matrix. In the output you should write the sparsity of the cut that you find and the id of the vertices in the smaller side of the cut. Please submit your code and the output to Canvas.

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-			A+	D+	B
Chris	B-		B+	C		B	B+
David	A+		B-	A-		A-	
Elise		B-	D+	B+		B	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.

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, $\{1, 2, \dots, n\}$; 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.