**CSE 521: Design and Analysis of Algorithms** **Fall 2018**

# Problem Set 4

*Deadline: Dec 5th in Canvas*

1) Let $G$ be a $d$-regular connected graph, and let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of the normalized Laplacian matrix of $G$, $\tilde{L}_G$. Show that $G$ is bipartite if and only if $\lambda_n = 2$.

   **Hint:** Recall that normalized Laplacian matrix of a $d$-regular graph is $\tilde{L} = L/d = (dI - A)/d = I - A/d$. Define $\tilde{M} = I + A/d$. Observe that $\tilde{M} = 2I - \tilde{L}$. Prove that if $\lambda$ is an eigenvalue of $\tilde{L}$ then $2 - \lambda$ is an eigenvalue of $\tilde{M}$. Therefore, 2 is the largest eigenvalue of $\tilde{L}$ if and only if 0 is the smallest eigenvalue of $\tilde{M}$. So, to prove the claim it is enough to show that $G$ is bipartite if and only if the smallest eigenvalue of $\tilde{M}$ is zero.

   Now, recall that for any vector $x$, $x^T \tilde{L} x = \sum_{i \sim j} \frac{1}{d}(x_i - x_j)^2$. Prove that for any vector $x$,

$$x^T \tilde{M} x = \sum_{i \sim j} \frac{1}{d}(x_i + x_j)^2.$$

   So, to prove the claim show that $G$ is bipartite if and only if $\min_x x^T \tilde{M} x = \min_x \sum_{i \sim j} \frac{1}{d}(x_i + x_j)^2 = 0$.

   **Extra Credit:** Prove the claim for non-regular graphs $G$.

2) We say a graph $G$ is an expander graph if the second eigenvalue of the normalized Laplacian matrix ($\tilde{L}_G$), $\lambda_2$ is at least a constant independent of the size of $G$. It follows by Cheeger's inequality that if $G$ is an expander, then $\phi(G) \geq \Omega(1)$ independent of the size of $G$. It turns out that many optimization problems are "easier" on expander graphs. In this problem we see that the maximum cut problem is easy in *strong* expander graphs. First, we explain the expander mixing lemma which asserts that expander graphs are very similar to complete graphs.

   **Theorem 4.1** (Expander Mixing Lemma)**.** *Let $G$ be a $d$-regular graph and $1 = \lambda_1 \geq \lambda_2 \geq \dots \lambda_n \geq -1$ be the eigenvalues of the normalized adjacency matrix of $G$, $A/d$. Let $\lambda^* = \max\{\lambda_2, |\lambda_n|\}$. Then, for any two disjoint sets $S, T \subseteq V$,*

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

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

   Use the above theorem to design an algorithm for the maximum cut problem that for any $d$ regular graph returns a set $T$ such that

$$|E(T, \overline{T})| \geq (1 - 4\lambda^*) \max_S |E(S, \overline{S})|.$$

   Note that the performance of such an algorithm may be terrible if $\lambda^* > 1/4$, but in strong expander graphs, we have $\lambda^* \ll 1$; for example, in Ramanujan graphs we have $\lambda^* \leq 2/\sqrt{d}$. So the number of edges cut by the algorithm is very close to optimal solution as $d \to \infty$. It turns out that in random graph $\lambda^* \leq 2/\sqrt{d}$ with high probability. So, it is easy to give a $1 + O(1/\sqrt{d})$ approximation algorithm for max cut in most graphs.

3) 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+ | | A- | B | B+ |
| David | A+ | | B- | A | | A- | |
| Elise | | B- | D+ | B+ | | B | C+ |

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

4) In the congestion minimization problem we are given a connected (undirected) graph $G = (V, E)$ and a set of pairs $s_i, t_i$ of vertices of $G$ for $1 \le i \le k$. We want to choose *exactly one* path between each pair $s_i, t_i$ ($k$ paths in total) such that for each edge $e \in G$, the number of paths that use $e$ is as small as possible. Consider the following LP-relaxation for this problem:

$$\begin{aligned} \min\ & z \\ \text{s.t.} \sum_{P: e \in P} f_P &\le z \qquad \forall e \\ \sum_{P \in \mathcal{P}_{s_i,t_i}} f_P &= 1 \qquad \forall i \\ f_P &\ge 0 \qquad \forall P. \end{aligned} \tag{4.1}$$

Here, $\mathcal{P}_{s_i,t_i}$ represent the set of all paths connection $s_i$ to $t_i$.

a) Prove that the above LP gives a relaxation of the problem

b) **Extra Credit:** Design an algorithm to *round* the solution to exactly one path connecting each $s_i$ to $t_i$.

c) **Extra Credit:** Prove that your algorithm gives an approximation factor of $O(\log n / \log\log n)$ to the problem.

5) **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 $L$, $\{1, 2, \ldots, n\}$; 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 $\delta$ 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 $\gamma n$.

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