CSE 521: Design and Analysis of Algorithms **Fall 2019**

Assignment 4

Deadline: Nov 25th (at 6:00PM) in Canvas

P1) Let $A \in \mathbb{R}^{m\times n}$ and suppose we want to estimate Av for a vector $v \in \mathbb{R}^n$. Here is a randomized algorithm for this task. Choose the i-th column of A, A_i, with probability

$$p_i = \frac{\|A_i\|^2}{\|A\|_F^2}$$

and let $X = A_i v_i / p_i$. Show that $\mathbb{E}[X] = Av$. Calculate $\text{Var}(X) = \mathbb{E}\left[\|X\|^2\right] - \|\mathbb{E}X\|^2$. Suppose, in preprocessing, we have calculated $p_1, \dots, p_n$ and we have made an array $b[]$ where $b[i] = p_1 + \dots + p_i$ for all $1 \le i \le n$. This should help you to generate a random column in time $O(\log n)$. Design an algorithm that for a given $\epsilon > 0$ and $v \in \mathbb{R}^n$ runs in time $O(m\frac{\log(n)\|A\|_F^2}{\epsilon^2\|A\|^2})$ and outputs a vector $y \in \mathbb{R}^m$ such that with probability at least 9/10

$$\|Av - y\| \le \epsilon \|A\| \|v\|$$

Note that if A has a few large eigenvalues and many small eigenvalues, then we can estimate Av in linear time as opposed to quadratic time using your algorithm.

P2) In this problem we prove that for any (undirected unweighted) graph G, $\chi(G) \le \lambda_1(A_G) + 1$, that is we can color vertices of G with (at most) $\lambda_1(A_G) + 1$ colors such that every adjacent pair of vertices have distinct colors where $\lambda_1(A_G)$ is the largest eigenvalue of the adjacency matrix of A. For example if G is a triangle,

$$A_G = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Then, $\lambda_1(G) = 2$ and we can color vertices of G with 3 colors, 1 with A, 2 with B and 3 with C.

a) Show that $\lambda_1(A_G)$ is at least the average degree of G, i.e., $\lambda_1(A_G) \ge \text{avg}_i d_i$.

b) For a set $S \subseteq V$ let $G[S]$ denote the induced subgraph of G on S, i.e., the graph with vertex set S where any pair is connected iff they are connected in G. Show that for any $S \subseteq V$, $\lambda_1(A_G)$ is at least the average degree of vertices of $G[S]$.

c) Design an algorithm that colors the vertices of G with at most $\lambda_1(A_G) + 1$ many colors.

P3) 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]. Note that to run a walk of length t on G, as described above, we only need $m + O(t)$ many random bits (assuming d is an absolute constant). Prove that for any integer $r < m$ we can improve the success probability to $1 - 1/2^r$ by running A only $O(r)$ many times using only $O(m)$ many bits. So, we need significantly less random bits compared to the simplest approach which uses $O(rm)$ many random bits.

P4) In this problem you are supposed to implement the spectral partitioning algorithm that we discussed in class. You are given a giant network, "com-DBLP" 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 1%. 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.

P5) **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 $|S| \leq \gamma n_L$,

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

[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

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.