Problem Set 2

- Due Date: **15 October 2023**
- The points for each problem is indicated on the side. The total for this set is **60** points.
- The problem set has a fair number of questions so please do not wait until close to the deadline to start on them. Try and do one question every couple of days.
- Turn in your problem sets electronically (PDF; either LaTeXed or scanned etc.) on Piazza.
- Collaboration is encouraged, but all writeups must be done individually and must include names of all collaborators.
- Referring to sources other than the text book and class notes is strongly discouraged. But if you do use an external source (eg., other text books, lecture notes, or any material available online), ACKNOWLEDGE all your sources (including collaborators) in your writeup. This will not affect your grades. However, not acknowledging will be treated as a serious case of academic dishonesty.
- Be clear in your writing.

---

1. **[The Affine Line graph]** (1 + 4 + 2 + 3)

   Let $\mathbb{F}$ be a finite field. Consider the following graph G whose vertex set is $\mathbb{F}^2$ and edges set E defined as

   $$E = \{((a,b),(c,d)) \ : \ ac = b + d\}\,.$$

   One way to interpret this is the point (a,b) is connected to all points (c,d) on the line $y = ax - b$.

   (a) Show that G is $|\mathbb{F}|$-regular.

   (b) Compute the adjacency matrix of the graph G^2. What are its eigenvalues?

   (c) Use the above to show that $\lambda(G) \leq \frac{1}{\sqrt{|\mathbb{F}|}}$.

   (d) Starting with this, and using the graph operations seen in class, show that you can construct a $(D^8, D, 1/8)$-spectral expander for some suitably large constant D.

2. **[Chernoff's bound for expander walks]** (3 + 3 + 3 + 6)

   In this problem, you will see a generic way to go from a *hitting-set tail* to a *Chernoff tail* due to Kabanets and Impagliazzo which we can instantiate for expander walks.

   Lemma 1. *Let $X_1, \ldots, X_t \in \{0,1\}$ be random variables (possibly correlated) such that for any subset $S \subseteq [t]$, we have*

   $$\Pr\left[\bigwedge_{i \in S} X_i = 1\right] \leq \mu^{|S|}.$$

   Then, for any $\varepsilon > 0$ with $\mu + \varepsilon < 1$, we also have

   $$\Pr\left[\frac{\sum X_i}{t} > \mu + \varepsilon\right] \leq e^{-\mathrm{KL}(\mu+\varepsilon \| \mu)\cdot t} \leq \exp(-\Omega(\varepsilon^2 t)).$$

The quantity $\mathrm{KL}(p\|q)$, called the Kullback-Leibler divergence, or relative entropy, equals $p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$. Standard Taylor arguments show that $\mathrm{KL}(\mu+\varepsilon\|\mu) = O(\varepsilon^2)$.

In this problem, you will prove the above lemma, and then instantiate it with expander random walks to get the upper-tail bound.

Let $p \in [0,1]$ be a parameter to be chosen shortly. Consider the following event where a set $S \subseteq [t]$ is chosen at random by adding each element $i \in [t]$ to S independently with probability p. Let M be the following expression:

$$M = \Pr_{S, X_1, \ldots, X_t} \left[\bigwedge_{i \in S} X_i = 1 \right]$$

(a) Show that $M \leq (p\mu + (1-p))^t$.

(b) Show that

$$\Pr_{S, X_1, \ldots, X_t} \left[\bigwedge_{i \in S} X_i = 1 \;\Big|\; \sum X_i > (\mu+\varepsilon)t \right] \geq (1-p)^{(1-\mu-\varepsilon)t}.$$

to conclude that

$$M \geq \Pr\left[\sum X_i > (\mu+\varepsilon) \cdot t \right] \cdot (1-p)^{(1-\mu-\varepsilon)t}.$$

(c) By setting $p = \frac{\varepsilon}{(\mu+\varepsilon)(1-\mu)}$, argue that

$$\begin{aligned} \Pr\left[\sum X_i > (\mu+\varepsilon) \cdot t \right] &\leq \left(\left(\frac{\mu}{\mu+\varepsilon} \right)^{\mu+\varepsilon} \cdot \left(\frac{1-\mu}{1-\mu-\varepsilon} \right)^{1-\mu-\varepsilon} \right)^t \\ &= e^{-\mathrm{KL}(\mu+\varepsilon\|\mu) \cdot t} \end{aligned}$$

(d) The above is a generic way to use a 'hitting probability' bound into a concentration bound. Instantiate this for expander random walks to prove the following result:

> Let G be an (N, D, λ)-expander. Suppose $B \subseteq [N]$ with $\mu = \frac{|B|}{N}$. Let $\mu' = \mu(1-\lambda) + \lambda$, and $\varepsilon > 0$ so that $\mu' + \varepsilon < 1$. If $v_1, \ldots, v_t$ is a random walk in G (pick v_1 uniformly at random, and keep choosing a uniformly random neighbour), then
>
> $$\Pr\left[\frac{|\{v_1, \ldots, v_t\} \cap B|}{|B|} > \mu' + \varepsilon \right] \leq \exp(-\Omega(\varepsilon^2 t)).$$

Although this problem worked with only indicator random variables (where $X_i = \mathbf{1}[v_i \in B]$), one can work with more general "weight" functions $f : V \to [-1, 1]$ and expander random walks give a pretty good estimate for $\mathbb{E}_v[f(v)]$. For more details see Thm 4.22 in Vadhan's manuscript.

3. **[Spectral gap of general regular graphs]** (1 + 2 + 2 + 4 + 4 + 2)

In this problem we will show that $\lambda(G) \leq 1 - \frac{1}{\mathrm{poly}(n,d)}$ for any d-regular n-vertex non-bipartite graph. In the process also learn about a very useful object called the Laplacian of a graph.

For an n-vertex d-regular undirected graph, define the *Laplacian* of the graph G (denoted by L_G) as

$$L_G = d \cdot I - A_G$$

where A_G is the adjacency matrix of G.

For a symmetric matrix M, we shall write $M \succeq 0$ to mean that $x^T M x \geq 0$ for all $x \in \mathbb{R}^n$. This is equivalent to stating that all eigenvalues of M are non-negative, and such matrices are also called positive semi-definite matrices (PSD) matrices.

We will extend this to a partial order between matrices to say that $A \succeq B$ if and only if $A - B \succeq 0$. If A and B are PSD matrices, this also implies that if $\lambda_1 \geq \ldots \geq \lambda_n$ and $\nu_1 \geq \ldots \geq \nu_n$ are the eigenvalues of A and B respectively, then $\lambda_i \leq \nu_i$ for all i.

(a) If H is a subgraph of G, show that $L_G \succeq L_H$.

(b) If K_n is the complete graph on n-vertices (without self-loops), what are the eigenvalues of L_{K_n}?

(c) Show that for any real numbers $x_1, \ldots, x_n$, we have

$$(n-1) \cdot \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \geq (x_1 - x_n)^2.$$

Use this to conclude that if P_n is a path graph consisting of edges $\{(1,2),(2,3),\ldots,(n-1,n)\}$ and $E_{1,n}$ is the graph with a single edge $(1,n)$, then

$$L_{P_n} \succeq \frac{1}{n-1} \cdot L_{E_{1,n}}.$$

(d) For a connected graph G on n-vertices, show that

$$\binom{n}{2} \cdot L_G \succeq \frac{1}{(n-1)} L_{K_n}$$

[Hint: For each pair of vertices (i,j), take the path $P_{i,j}$ in the graph G that connects i to j. Instantiate the previous subdivision for this path and sum up over all such pairs $(i,j) \in \binom{[n]}{2}$.]

(e) Show that if G is a connected graph, then all nonzero eigenvalues of L_G are at least $1/n^2$. In particular, if G is a connected d-regular graph, then all non-trivial eigenvalues of G are at most $(1 - \frac{1}{dn^2})$.

(This doesn't quite give us a spectral gap yet, we are not claiming that the absolute values of eigenvalues is bounded away from 1, but the next subdivision addresses this.)

(f) Let G be an n-vertex d-regular non-bipartite graph. Show that

$$\lambda(G) \leq 1 - \Omega\left(\frac{1}{d^2 n^2}\right)$$

[Hint: Consider $G' = G^2$, which must be connected and apply the previous argument.]

One of the exercises in Vadhan's notes is a far shorter way to get the above spectral gap bound, but we felt that perhaps the above route shines more light on some of the steps.

4. **[An optimal non-averaging sampler]** (5 + 10)

Suppose $f : \{0,1\}^m \to \{0,1\}$ is some function and $\mu = \mathbb{E}_x[f(x)]$. A (δ, ε)-*sampler* is a randomized algorithm that queries f at various points and outputs some estimate $\hat{\mu}$ with the property that

$$\Pr[|\hat{\mu} - \mu| > \varepsilon] \leq \delta.$$

We are primarily interested in two parameters of such samplers — how many queries did it make, and how many random bits did it use. For this entire problem, assume that we have a *strongly-explicit* $(2^m, d, 0.5)$-spectral expander for some constant d.

(a) Using expanders, show how one can obtain a (δ, ε)-sampler that makes at most $O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ queries and uses at most

$$m + O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right) \text{ random bits.}$$

(You may assume Theorem 4.22 from Vadhan's manuscript, which is a stronger form of the Question 2 in this set, for this problem. You may also assume that there are strongly explicit constant-degree expanders on 2^m vertices.)

(b) Suppose we have a $((1/8), \varepsilon)$-sampler $\mathcal{S}$ that makes Q queries and uses R random bits. In the last problem set, you studied the "median of averages sampler" built from $\mathcal{S}$:

> Run the sampler $\mathcal{S}$ for t independent trials to obtain $\hat{\mu_1}, \ldots, \hat{\mu_t}$. Output the *median* of these estimates.

You should have shown that this new sampler will be an (δ, ε)-sampler if $t = O\left(\log \frac{1}{\delta}\right)$ and instantiated $\mathcal{S}$ using pairwise independence.

Use expander random walks to now construct an (δ, ε)-sampler that makes at most $O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ queries and uses at most

$$O\left(m + \log \frac{1}{\varepsilon} + \log \frac{1}{\delta}\right) \text{ random bits.}$$

You have now seen a sampler that has *optimal* number of queries and random bits used (up to constants) but is **NOT** an averaging sampler! Obtaining an averaging sampler with the same performance is an open problem.

5. **[Error reduction for randomised algorithms]** (5)

Suppose you have a randomised algorithm $\mathcal{M}$ for some language L. Let's say that on inputs of length n, the algorithm tosses $m(n)$ random coins and runs for time $t(n)$ and we have the guarantee that probability of error is at most $1/3$. That is,

$$\begin{aligned} x \in L &\implies \Pr_{r \in \{0,1\}^m}[\mathcal{M}(x, r) = 1] \geq 2/3, \\ x \notin L &\implies \Pr_{r \in \{0,1\}^m}[\mathcal{M}(x, r) = 1] \leq 1/3. \end{aligned}$$

However, you wish to have the probability of error no more than δ. Based on what you have seen in class so far, how would you modify the above algorithm to drive the probability of error down to δ?

How much time does your modified algorithm take? How many random bits does your modified algorithm use?

(This question is purposefully vague as we will be revisiting this question multiple times.)
