

18.409: Problem Set 1

Jonathan Kelner

September 30, 2007

How this Problem Set Works

The problems on this problem set vary fairly widely in difficulty. In order to allow people who want hard problems to be able to have them without preventing people who would like more straightforward problems from being able to do them in a reasonable amount of time, I have provided hints online for many of the problems. Without the hints, some (but not all) of these problems will be quite difficult. With them, I hope that they are more manageable. Use as many of the hints as you like; just note in your solution whether you are doing so.

Also, recall that you need not do all of the problems on the problem set. Only do as many as is helpful for you to learn the material.

Finally, many of these problems have solutions that can easily be found by web searching. Please don't do that. It's not very nice.

Problem 1: Bipartite Graphs

Let G be a graph, let A be its adjacency matrix, let D be its degree matrix, and let $W = AD^{-1}$ be its random walk matrix.

- (a) Show that if G is bipartite then there is some eigenvector v of W with eigenvalue -1 .
- (b) Show that if G is connected and has an eigenvector of eigenvalue -1 , then G is bipartite.

Problem 2: Matrix Inequalities and Lower Bounds for λ_2

In this problem, we shall develop a technique for proving lower bounds on λ_2 that are often much tighter than those coming from Cheeger's inequality.

(a) Let A and B be real symmetric matrices with respective eigenvalues $\lambda_1 \leq \dots \leq \lambda_n$ and $\mu_1 \leq \dots \leq \mu_n$. Use the Courant-Fischer theorem to show that if $A - B$ is positive semidefinite, then $\lambda_1 \geq \mu_1$, $\lambda_2 \geq \mu_2$, and $\lambda_n \geq \mu_n$. (In fact, $\lambda_k \geq \mu_k$ for all k by the same argument, but we only stated the Courant-Fischer theorem in class for λ_1 , λ_2 , and λ_n .)

(b) Let L_{P_n} be the Laplacian of the path graph on n vertices (and $n - 1$ edges), and let $L_{i,j}$ be the Laplacian of the graph consisting of a single edge from vertex i to vertex j . Show that

$$(n - 1)L_{P_n} - L_{1,n}$$

is positive semidefinite. **[Hint available online.]**

(c) We'll now use what we've developed above to prove a lower bound on the second eigenvalue of the path graph. This will match the $O(1/n^2)$ upper bound we proved in class.

To simplify the manipulations, we will use the notation $A \succeq B$ to indicate that $A - B$ is positive semidefinite. For example, the result of the previous part may be written as

$$(n-1)L_{P_n} = (n-1) \sum_{i=1}^{n-1} L_{i,i+1} \succeq L_{1,n}.$$

We will embed the complete graph K_n in the path graph by mapping the i^{th} vertex of K_n to the i^{th} vertex of P_n . This maps each edge of the complete graph to a sequence of edges in the path graph. Use the inequality from the previous step and this embedding to show that

$$\lambda_2(P_n) \geq \Theta(1/n^2).$$

[Hint available online.]

(d) Embed the complete graph on $n = 2^k - 1$ vertices into the complete binary tree T_n on the same number of vertices. Use this to show that the second eigenvalue of T_n is at least $\Omega(1/n \log n)$. [Hint available online.]

Problem 3: The Efficacy of Spectral Partitioning

In class, we showed that the eigenvector v_2 corresponding to the second eigenvalue λ_2 of the Laplacian can be used to find a good cut in a graph. We proved the inequalities

$$\Theta(1) \frac{\phi^2}{d_{\max}} \leq \lambda_2 \leq \Theta(1)\phi.$$

We noted that both of these inequalities can be asymptotically tight, as evidenced by the path graph, which has $\phi = O(1/n)$ and $\lambda_2 = O(1/n^2)$, and the complete binary tree, which has $\phi = O(1/n)$ and $\lambda_2 = O(1/n)$.

Our proof of the first inequality (Cheeger's inequality) was constructive. We showed that the ratio ϕ_{spectral} of a cut obtained from v_2 obeys

$$\Theta(1) \frac{\phi_{\text{spectral}}^2}{d_{\max}} \leq \lambda_2 \leq \Theta(1)\phi.$$

In graphs where Cheeger's inequality is tight, such as the path graph, this gives us a cut of asymptotically optimal ratio. However, for the complete binary tree, this only guarantees a cut of ratio $O(\sqrt{n})$, even though a cut of ratio $O(1/n)$ exists. This seemingly gives us an example where spectral partitioning produces a significantly suboptimal result.

However, let's look at what happens when we run spectral partitioning on the complete binary tree. In the picture below, we have laid out the binary tree with x -coordinates given by the second eigenvector (and y -coordinates constructed to give a nice picture). If we use this eigenvector to obtain an embedding onto the real line and then cut the line at 0, we actually obtain the optimal cut of ratio $O(1/n)$. Thus, while Cheeger's inequality doesn't guarantee it, spectral partitioning still produces a good cut on the complete binary tree.

In this question, I ask you to construct a graph for which spectral partitioning *doesn't* work. That is, construct a graph G in which no cut produced using v_2 has a ratio better than $\Theta(\sqrt{\phi(G)d_{\max}})$.

Caveat: A poll of spectral graph theorists led me to conclude that this question is way too hard to put on a problem set without a decent hint. Nevertheless, I think that trying to construct such an example without being told what to construct is a fantastic way (perhaps the best that I know) to get a feel for how cuts relate to eigenvalues. As such, I've put a hint online, but I strongly encourage you to think about the problem a bit before looking at it.

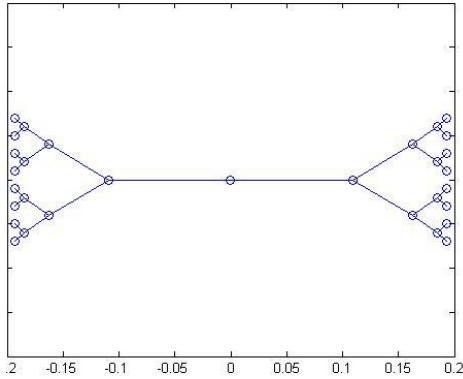


Figure 1: A binary tree with x -coordinate given by v_2

Problem 4: Proving the Chernoff Bound

For those of you who have not used the Chernoff bound much before, it might be helpful to work through a proof and understand why it holds. That is the goal of this problem.

(a) Prove **Markov's inequality**: If X is a *nonnegative* real random variable, then

$$\Pr[X \geq \lambda] \leq \frac{\mathbf{E}[X]}{\lambda}.$$

(b) Let X_i be a $\{0, 1\}$ random variable that is 1 with probability p_i and 0 with probability $1 - p_i$. Let

$$X = \sum_{i=1}^n X_i$$

be a sum of n of the X_i , and assume that the X_i are *independent*. (Note: without this assumption, the Chernoff bound is no longer true.) Let $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Show that for any t ,

$$\Pr[X \geq (1 + \epsilon)\mu] \leq \frac{\prod_{i=1}^n p_i e^{t p_i} + (1 - p_i) \cdot 1}{e^{(1+\epsilon)t\mu}}.$$

You may use the following fact without proof:

Fact. If X_1, \dots, X_n are *independent* random variables,

$$\mathbf{E} \left[\prod_{i=1}^n X_i \right] = \prod_{i=1}^n \mathbf{E}[X_i].$$

[Hint available online.]

(c) Show that for any $x \in \mathbb{R}$,

$$1 + x \leq e^x.$$

Apply this to part (b) to obtain the inequality:

$$\Pr[X \geq (1 + \epsilon)\mu] \leq \frac{e^{(e^t - 1)\mu}}{e^{(1+\epsilon)t\mu}}.$$

(d) Plug the value $t = \ln(1 + \epsilon)$ into the result of part (c). Use this to conclude that for any $\epsilon \in [0, 1]$,

$$\Pr[X \geq (1 + \epsilon)\mu] \leq e^{-\epsilon^2\mu/3}.$$

You may use the following fact without proof:

Fact. For any $\epsilon \in [0, 1]$,

$$(1 + \epsilon)\ln(1 + \epsilon) \geq \epsilon + \epsilon^2/3.$$

Problem 5: Existence of Bipartite Expanders

In this problem, we will prove that bipartite expanders exist. We will do so using the probabilistic method. That is, we will show that a random bipartite graph chosen from a certain distribution is an expander with nonnegligible probability. This will show, in particular, that expanders exist.

We will show:

Theorem 1. *Construct a random left d -regular bipartite graph G with N left vertices and N right vertices as follows. For each vertex on the left, choose d (not necessarily distinct) neighbors on the right uniformly at random. For any d , there exists an $\alpha = \alpha(d)$ such that, with probability at least $1/2$, G is an $(\alpha, d - 2)$ -expander.*

(a) Let S be a set of left vertices of size k , and let $N(S)$ be its neighbors. Argue that the probability that $N(S) \leq (d - 2)k$ is at most

$$\binom{kd}{2k} \left(\frac{kd}{N}\right)^{2k}.$$

(b) Show that the probability that there exists *any* set of size k such that $N(S) \leq (d - 2)k$ is at most

$$\left(\frac{cd^4k}{N}\right)^k.$$

for some constant c . Conclude that there is some $\alpha = \alpha(d)$ such that, for any $k \leq \alpha N$, this probability is at most 4^{-k} .

You may use the following inequality without proof:

Fact.

$$\binom{a}{b} \leq \left(\frac{ae}{b}\right)^b.$$

(c) Use this to prove Theorem 1.

Problem 6: Expander Codes

This problem is for people who are familiar with the basic definitions from coding theory (or who are willing to look them up). The goal of the problem will be to construct an asymptotically good code of constant rate that can be decoded in linear time.

Let G be a bipartite graph with N vertices on the left and $N/2$ vertices on the right with no multiple edges or self-loops such that:

1. All vertices on the left have degree d
2. All vertices on the right have degree $2d$
3. Every set S on the left of size at most αN has $|N(S)| > \frac{3d}{4}|S|$ for some constant $\alpha > 0$.

We'll define a code using this graph. In this problem, we will not worry about the encoding step; we will merely define which bit strings correspond to codewords and study a decoding algorithm. We'll see how to do this using expanders.

Label the vertices on the left with the indices $1, \dots, N$, and associate a bit b_i to each such vertex. Each vertex r on the right will give a constraint on these bits as follows: we shall require that the sum modulo 2 of the bits assigned to r 's neighbors must be 0. That is,

$$\sum_{(i,r) \in E} b_i \equiv 0 \quad (2).$$

The codewords of our code will consist of all N -bit strings that satisfy all $N/2$ constraints.

General hint for this problem: Analyzing these codes revolves around a very similar technique to the one we used in class for the distributed algorithm for nonblocking flows: examining and counting vertices on the right with unique neighbors in some set of vertices on the left. This will be useful in both parts (a) and (c).

- (a) Show that any two codewords differ in at least αN coordinates. **[Hint available online.]**
- (b) Suppose we are given a string s that is of distance at most $\alpha N/2$ from some codeword c . Consider the following iterative decoding algorithm to recover the codeword c :

Find a vertex v on the left with more than half of its neighbors unsatisfied, and flip the corresponding bit of s . Repeat until there are no longer any such vertices.

Show that this algorithm stops (not necessarily with the right answer) after at most $N/2$ iterations. You should not need to use expansion to prove this.

Remark 2. It is not very difficult (under the proper model of what it means to be linear time) to implement the above procedure in linear time if you can give a linear bound on the number of iterations, so this guarantees that the algorithm runs in linear time. Note, however, that it doesn't yet guarantee that we've found the right answer; it just guarantees that the algorithm terminates.

- (c) Show that if you start with a string that is at most $\alpha N/2$ from some codeword C then the decoding algorithm never, at any iteration, produces a codeword that is more than αN away from C . Use this to conclude that the algorithm terminates with the correct value. **[Hint available online.]**