

Problem Set 5

1 Introduction to Optional Problem Set

This problem set is optional. You can do some problems from it if you need some extra credit. It must be turned in either by email or to Daniel Spielman's mailbox on the first floor of AKW by 4:00PM on Friday, December 10th.

There are two sections to this problem set. The first consists of theoretical problems. The second contains the experimental problems.

2 Homework Policy

You may discuss the problems with other students. But, you must write your solutions independently, drawing on your own understanding. You should cite any sources that you use on the problem sets other than the textbook, TA and instructor. This means that you should list your collaborators.

If you are doing the experimental problems, it goes without saying that you should write your own code.

You **may not** search the web for solutions to similar problems given out in other classes. If you think this policy needs any clarification, please let me know.

3 Theoretical Problems

1. Consider the following variant of the Gossip problem. We begin with n vertices, the i th of which starts with value x_i . Whenever vertices i and j communicate, they average their two values. At each time step, some pair of vertices will communicate. You may assume that there is some number T so that in every interval of T consecutive time steps, the graph containing the edges that have communicated during those steps is connected. Prove that the values contained at the vertices eventually converges.

That is, for every $\epsilon > 0$, prove that for every i and j , after enough steps, we will have

$$|x_i - x_j| < \epsilon.$$

2. I'd like to give you a problem about what happens to a graph when you repeatedly use spectral partitioning to divide it into pieces. But, the problem was too difficult. So, I'll give you a simpler problem.
 - a. Let G be a graph. Fix a number θ less than $1/2$, and let S be the largest set of vertices in G with $d(S) \leq d(V)/2$ and $\phi(S) \leq \theta$. Prove that if $d(S) \leq d(V)/4$, then $\phi_{G(V-S)} \geq \theta/10$.
 - b. In this problem, we will analyze what happens when we repeatedly divide a graph according to its cuts of minimum conductance. While we cannot necessarily compute these efficiently, it is still interesting to see what happens.

I will now describe the process more concretely. We begin by choosing some target conductance, θ , to be chosen later. Given a graph G , choose some set S of minimum conductance. If the set S has conductance less than 10θ , then divide the vertex set of G into two pieces: S and $V - S$. Now, recurse on the induced subgraphs $G(S)$ and $G(V - S)$. Keep going until you have partitioned the vertices into subsets S_1, \dots, S_k so that each induced subgraph $G(S_i)$ has conductance at least θ .

Prove that there exists a constant α so that if we set $\theta = \alpha/\log_2 d(V)$, then when we have finished this process at least half of the edges of G lie inside the induced subgraphs $G(S_1), \dots, G(S_k)$. That is, less than half of the edges have been cut by this process.

4 Experiments

In these experiments, you will explore spectral partitioning. To do this, you will need to compute eigenvectors of graphs. As part of your write-up, you must provide evidence that you are computing these correctly. As before, you should submit all of your code (printed), along with a clear explanation of what you did in the experiments, and what your code is doing.

1. In Lecture 19, I mentioned one approach to spectral clustering: to divide a graph into k pieces, we take the first $k - 1$ non-trivial eigenvectors and then run k -means on the embedding that they provide.

In Lecture 20, I alluded to another approach. In that approach, one uses just one eigenvector, and repeatedly cuts the graph in half, until one obtains k pieces. That is, one should first compute an eigenvector of the walk matrix (or equivalently the normalized Laplacian) and take **the best** Cheeger cut of it. That is, take the prefix of values that minimizes the conductance of the cut. This then divides the graph in half. Once you have two subgraphs, you should then consider applying this operation in both, but only apply the operation that minimizes the normalized cut measure. Keep doing this until you have divided the graph into k pieces. That is, at each step you greedily divide the component that will minimize the normalized cut score.

Try both of these operations on your graph, for all values of k between 2 and 10. As the variant involving k -means is randomized, repeat this experiment at least 10 times and take the best result. For each value of k , compute and report the normalized cut scores obtained by both approaches. Which is better?