CS 292F: Project Proposal

Nathan Wachholz

May 4, 2022

I am currently doing research in computational geometry with Subhash Suri. Therefore, I'm interested in how graph Laplacians, and other topics we've discussed, relate to geometric questions. The most applicable topic (from Spielman's book, and a bit of Googling) seems to be graph partitioning (as a means of clustering).

The geometric k-clustering problem is: Given a set of n points in some metric space (doesn't need to be euclidean), we want to cluster these points into k groups, minimizing some property (such as max distance between any two points in a group). This optimization problem is NP-Hard in general, so we have developed a number of approximation methods. I wonder if graph partitioning of geometric graphs can lead to good approximations. There are a few things to investigate.

First: the type of input graph. Obviously, we represent points as vertices, and edges as the distances between them. But what edges should we include? To start with, we can calculate a complete geometric graph, with the distances between every $O(n^2)$ pair of points. We can then consider approximations to this graph, such as k-spanners, triangulations, and spanning trees. These only have $O(n)$ edges, making sparse representation viable. How does our ability to partition points vary across these inputs?

Second: how to perform and interpret graph partitioning. The graph partitioning method we discussed in class involves first calculating the Fiedler value/vector. Then we partition the points into two groups depending on if their entry in the Fiedler vector is above or below the *median* entry. Can we look at quartiles, or otherwise further group the Fiedler vector entries? What if we allow our partitions to have different sizes? Optimal 1D k-clustering is solveable in polynomial time—would clustering by Fiedler entries lead to a useful partitioning in higher dimensions?

Here is an outline of what I am planning to do:

- Computational experimentation: look at Laplacians of complete geometric graphs. Can I find any interesting properties of these matrices?
- Alternate metrics: if we use metrics other than euclidean distance (like L_∞ or L_1). Do any properties carry over or emerge?
- Sparse approximations: consider approximations of geometric graphs. Do any properties carry over?
- Partitioning and clustering: partition points using geometric graphs. How can I cluster points from graph partitions? How does approximating the input graph change the clustering? Can I make any statements about how good these clusters approximate the optimal?

I've already spent some time looking for literature on this—without much success. I'll look a bit more, but imagine that most of my work will be playing around in Matlab. Please let me know if you would prefer something more strictly focused on literature/surveying, or focused on computation.