Tanya Jeffries

CS 665: Assignment 4 (Research Project Proposal)

For my research project I plan on employing techniques similar to the PCA-based algorithm, HDE (High Dimensional Embedding) [1], to bipartite graph drawing. To my knowledge PCA has never been applied to these types of problems before, so this project will serve as an initial experiment to see how well PCA does in the bipartite graph drawing context.

The algorithm that I am using as a reference is Yehuda Koren's HDE algorithm for drawing general graphs (i.e. it does not assume any particular structure for input graphs)[1]. The algorithm works by selecting m special "axis" nodes in a greedy fashion according to graph distances, where m is a parameter decided beforehand (in their paper that deals with general graphs on the order of 10^5 nodes they recommend a value of m = 50). After selecting said axis nodes, graph theoretic distances from all nodes in the graph to each of the axis nodes is calculated. Principal component analysis is then applied to the resulting distance-based matrix and the layout of the graph is calculated in two dimensions.

Bipartite graph drawing is typically done in a layered fashion with vertices of the partitioning subsets being placed on horizontal parallel lines /layers (one for each vertex subset) with edges running in between. Within this framework there are two problems, one-sided crossing minimization (OSCM) and two-sided crossing minimization (TSCM). In OSCM the coordinates of nodes in one of the layers are known and fixed, while the nodes in the opposing layer are free to permute, with the goal being to minimize the total number of edge crossings. In TSCM the nodes in both layers are free to permute, with the same goal of minimizing total edge crossings. Both OSCM and TSCM have been proven to be computationally hard, so efficient and practical methods are sought to solve these problems.

My past and current research work has been on addressing OSCM in unweighted and weighted bipartite graph drawing. To apply Yehuda's PCA-based techniques to OSCM, I was planning on using the nodes of the fixed layer as the axis/reference nodes. Graph theoretic distances of the free nodes from the fixed nodes would be calculated and used to form the distance matrix for PCA. Nodes in the free layer would then be ordered according to their coordinates designated by the eigenvector of the covariance matrix.

Since I already have several OSCM algorithms implemented, my plan is to test this PCA-based method against them on random unweighted bipartite graphs. To evaluate all of the algorithms I will be calculating their average crossings, runtime (in seconds), and percentage deviation from a lower bound. If the performance on the unweighted graphs is promising, I may also evaluate it against my algorithms on random weighted bipartite graphs. I will know that the technique is good if it results in average crossing counts that are comparable to my top-performing algorithms. By the time that my final report is submitted I plan on presenting a complete set of comparative test results for random bipartite graphs of varying size (probably in the range of 20 to 40 nodes per layer).

As PCA has never been applied to bipartite graph drawing before, I don't think that I can really say what a success would be. Certainly if it proves to be a promising technique for OSCM,

then that would be successful, but at the same time just experimenting with a new method and seeing how it turns out also answers a lot of questions.

References:

[1] Harel, David, and Yehuda Koren. "Graph drawing by high-dimensional embedding." *International Symposium on Graph Drawing*. Springer Berlin Heidelberg, 2002.