Prof. Dr. Ingo Scholtes
Chair of Informatics XV
University of Würzburg

Exercise Sheet 10

Published: January 27, 2023
Due: February 2, 2023
Total points: 10

This is the final exercise sheet for this semester

1. **Algebraic Connectivity and Spectral Clustering**

   (a) Show that the eigenvalue sequence of a Laplacian matrix $\mathcal{L}(G)$ of a network G with k connected components has k zeros. [2P]

   (b) Consider a simple model for networks consisting of two equally-sized clusters, where each cluster is a random Erdös-Renyi network and the two clusters are connected by "crossing" a configurable number of random links drawn from both clusters. We have implemented this model in a practice session accompanying lecture 12. Study for which value of the randomly crossed link pairs the cluster structures can be reliably detected based on the simple spectral clustering algorithm implemented in the practice session. [3P]

2. **Line Graphs and Laplacian Embedding**

   (a) Arc diagrams are a simple method to visualize networks. In an arc diagram, we places the nodes on a (virtual) horizontal line and connect them with semicircles that can be drawn above or below the line. Like most network visualizations, arc diagrams are more intelligible when there are fewer crossings between the lines that depict edges. A (somewhat naive) strategy to minimize crossings is to place nodes that are connected by an edge closer to each other. The cumulative distance between connected nodes can be quantified by the sum of squared distances of all node pairs connected by an edge, which we denote as Δ^2. [3P]

   We can now consider visualization algorithms that minimize this distance. Without further constraints the minimization of Δ^2 is subject to trivial solutions: First, we can place all nodes at the same coordinate, which trivially minimizes Δ^2. Similarly, we can trivially improve any assignment of nodes to (one-dimensional) coordinates $\mathbf{x}$ by simply rescaling them to $\alpha\mathbf{x}$ ($\alpha < 1$). To avoid this we thus impose the constraints that (i) the coordinates x_i of nodes i cannot be the same, and (ii) the assignment $\mathbf{x}$ has a fixed scale, e.g. $\|\mathbf{x}\| = 1$.

   Given a network with adjacency matrix $\mathbf{A}$ show that, subject to the constraints above, the vertex positions that minimize Δ^2 are given by $\arg\min_{\mathbf{x}\in\{\mathbf{x}:\mathbf{x}\perp\mathbf{1}\,\wedge\|\mathbf{x}\|=1\}} \mathbf{x}^{\mathbf{T}}\mathcal{L}\mathbf{x}$ where $\mathcal{L}$ is the Laplacian matrix.

   (b) The method explained in the previous task generates a so-called "embedding" of a network in a one-dimensional space. This method is not limited to embedding networks on a line. It can also be used to embed a network in a space of any dimensionality $0 < d \leq rank(\mathcal{L})$. Use the eigenvectors of the two smallest non-zero eigenvalues of the Laplacian matrix as coordinates to embed nodes in the following network topologies in a two-dimensional Euclidean space: [2P]

   - a two-dimensional grid lattice network
   - a three-dimensional cube lattice network
   - an Erős-Rényi network
   - a small empirical network from the `konect` database

   Compare these visualizations with the ones returned by `pathpy`'s force-directed layout. Discuss the performance of the eigenvector-based strategy in the empirical network.