Spectral Graph theory applied to Trade Networks

Anup Rao

October 1, 2018

This document gives some examples generated from simulations that show how ideas related to spectral graph theory might be used to analyze trade networks arising from real economies. My paper at `https://arxiv.org/abs/1702.03290` presents some of the math underlying the methods I am using here. The programs I wrote should scale to huge networks given a computer with enough memory.

Given an arbitrary network, spectral graph theory allows you to find a *nice* placement of all the nodes of the network in space. What *nice* means is hard to describe succinctly, but a good intuition is to think of what happens when every pair of nodes that is connected strongly attracts each other, and every pair of nodes that is disconnected weakly repels each other. The system will settle into some final state, and you could think of the drawing we obtain as the final state of the system under those rules. This intuition is not entirely accurate, but it is close enough. All of the examples I will discuss here have transactions of value 1, but everything can easily be generalized to handle different transaction weights.

Crucially, the algorithm that is placing the nodes knows *nothing* about the nodes other than how they are connected to other nodes. So, the drawings allow us to *find* important features of the network that we may not have known existed ahead of time.

Alan suggested trying to classify the nodes by industry. I give some examples of how one might do this at the end of the document. I do think there are many other kinds of things we could do with the data. I've given many examples to illustrate what is possible.

Examples

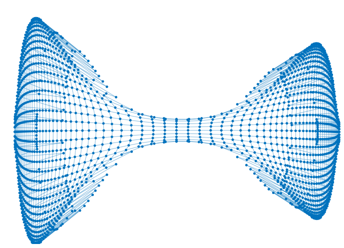

Figure 1: The dumbell network

In the dumbell network, shown in Figure 1, there is a narrow band of nodes that is the only connection between two large parts of the network. The drawing in the figure was produced by the theory, and it allows you to identify the few nodes that form the connection between the two large parts of the network. In contrast, if you used the method of eigenvalue centrality, you would find that almost all of the nodes in this picture are equally central. This is because the eigenvector corresponding to the largest eigenvalue of the adjacency matrix has entries that depend only on the number of nodes that each node is connected to, rather than the location of the node in the

What I am doing in this document is looking at *other* eigenvalues of a different matrix (the normalized Laplacian) instead.

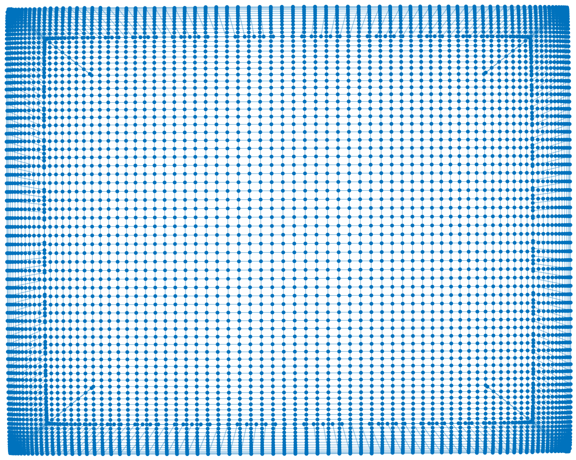

Figure 2: The grid network. This is an 80 by 80 grid.

entire network.

The next example is the grid network, shown in Figure 2. The drawing looks more or less how you might expect. (The edges are folded-in, which has to do with the way the math actually works.) This is actually a 2d picture of a 3d object. Another view of the same placement of nodes from a different direction can be seen in Figure 3.

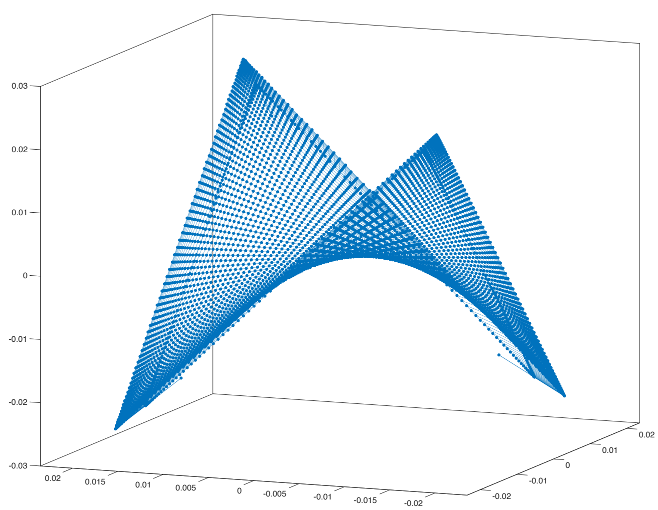

Figure 3: This is the same grid as in Figure 2, viewed from a different angle.

Suppose there was some obstruction that made it hard for the nodes whose x-coordinates were close to 40 to trade with their neighbors. I modeled this by taking the two middle columns of grids and deleting each node in this region with probability $2/3$. The result is shown in Figure 4. The picture is a little deceptive—it seems like the network has a lot fewer nodes, but it is actually only missing nodes from two columns in the middle. The rest of the nodes are bunched up on the sides.

You can use the picture to partition the network into the two parts that have few connections between them. If we wanted to continue analyzing each part, we could repeat the analysis on each side to identify structures within each side.

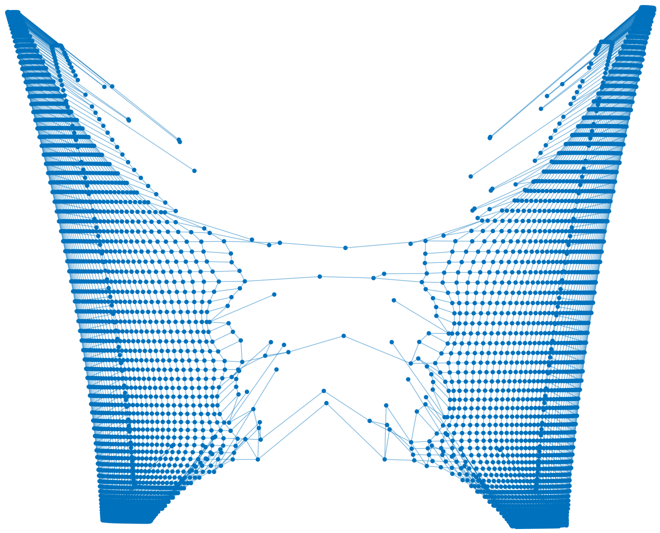

Figure 4: This is the grid with nodes in the middle two columns deleted with probability 2/3.

Now, let us see what happens if instead of decreasing the connectivity in a region, we increased it. Suppose someone were to build a road in this economy. One would expect that participants along the road would trade with others at larger distance along the road. To play with this, I picked a narrow strip of nodes and connected each node from the strip to 3 others from the strip, and drew the picture again. The results are visible in Figure 5. You can see that it is easy to identify the nodes near the road from the picture.

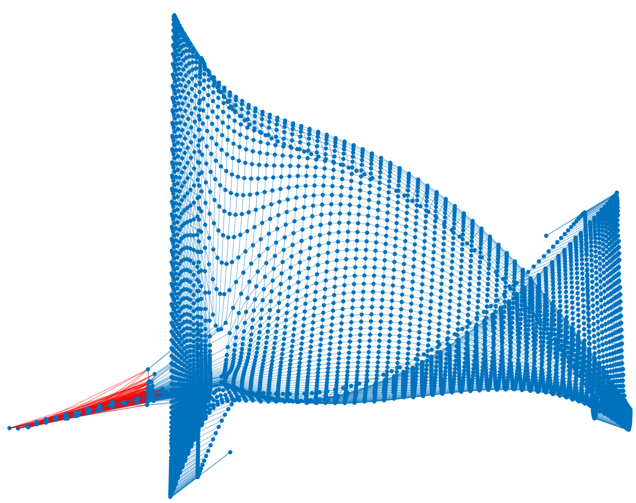

Figure 5: The grid network with an added *road*. Nodes near the road trade with additional nodes that are randomly chosen from the road. The added transactions are shown in red.

Next, in Figure 6, I modified the grid by taking three distinct circular regions where I connected each node to a few other nodes chosen randomly from within the boundary of the circle. I think of these circles again as high areas of activity. In addition, I sprinkled in many other random nodes throughout the network that connect to nearby nodes at a higher frequency. I hope it is clear that you can pick out these three regions from the picture, and the additional scattered nodes with higher activity do not really interfere with the picture.

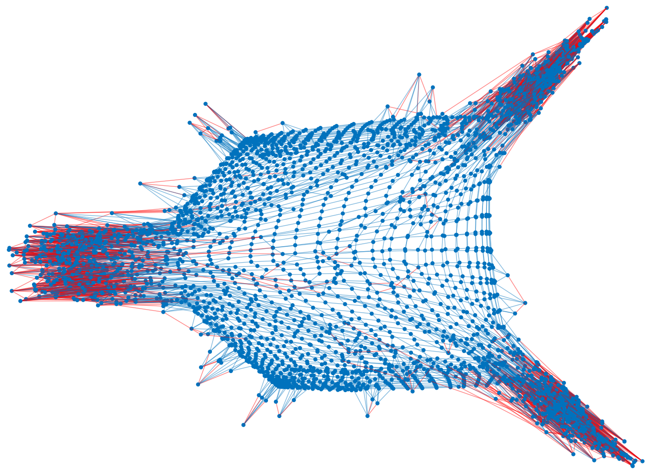

Figure 6: The grid network with three regions of increased activity, as well as several randomly scattered nodes of increased activity.

Alan suggested trying to classify the nodes in the network according to industry. To come up with a way to do this, I used a different process from what I used in the earlier examples. This time, the appropriate intuition is that nodes that are connected *repel* each other strongly and nodes that are disconnected weakly attract each other.

Figure 7 shows the results when the network is generated by adding 200 nodes in each of 5 different industry. Industry 1 sells to industry 2, which sells to industry 3 and so on. Between any two industries that sell to each other, every edge is included randomly with probability 1/5. You see that one can reconstruct the industries from the picture. In the final example, I added 500 random links to

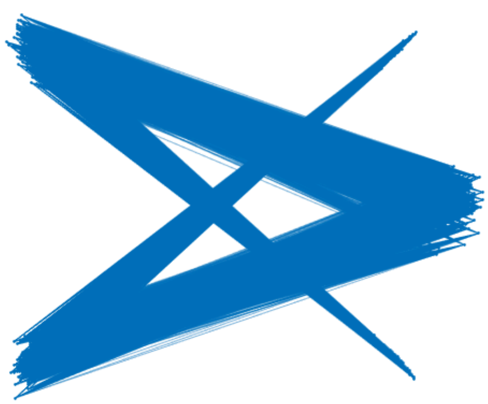

Figure 7: The industry network, plotted by making nodes repel each other along links. Here there are 5 industries in a linear supply chain.

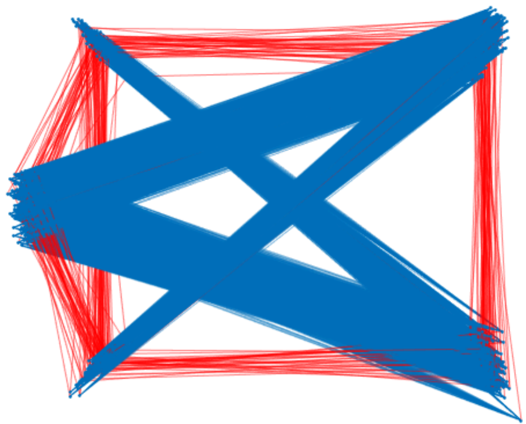

Figure 8: This is the same as Figure 7, except that 500 additional edges were randomly added to the network. The result seems to have flipped itself around, but it is still clear how to classify all the nodes by industry.

the industry network. You see that this did not really affect the placement of nodes by much. It seems to have flipped the figure around, but the classification remains clear.