Fun and Games with BTER Graph Generator

Ben Bales

June 10, 2014

1 Introduction

BTER [3] is a graph generator that models social network graphs my matching the degree and clustering coefficient distributions. It is hoped that by matching these distributions that the randomly generated graphs are enough like the original that computations on them are roughly computationally equivalent to computations on the original. In this study, eigenvalues and eigenvectors between the original and resultant graphs are compared to test their equivalency.

BTER works by creating a small set of tightly integrated subgraphs to match degree distributions and clustering coefficients and then connecting them in a global structure to create a single graph.

1. Building the small clusters is a three step process
   (a) Create a sorted list of vertices labeled with degrees from the original graph
   (b) Pop off vertices from the top of this list in size equal to the degree of the first vertex (so if the first degree is 3, then a group of 3 vertices is lumped off)
   (c) Interconnect these vertices probabilistically with a distribution that is a function of the graph type (set to a sort of decaying log form in the reference implementation)
2. Interconnecting all the small graphs is easy. For each cluster, use any unmet degrees (not all the degree parameters of each vertex are necessarily met in the prior step) from within the cluster to randomly link it to things across the graph

Again, BTER was designed to work specifically on social networks, but the idea seems robust enough it could be used for other graph types. This paper is going to test how effectively BTER can model a wide range of Laplacian graphs sourced from the Florida Matrix Archive [1]. The BTER implementation used is the BTER test implementation[4], so there is probably some room for improvement on the general problems tested in this study.

2 Methods

A significant number of mostly symmetric (all but one) graphs were selected from the archive. Every graph was turned into a symmetric Laplacian by first making all off diagonal elements symmetric (by assigning $A = A + A^T$), turning all the nonzero entries into negative ones, and then replacing the diagonal elements with the absolute value of the sum of the non-diagonal elements in each row (this is the lapacian command from Gilbert [2]).

Graphs were categorized into rough bins:

1. Social network graphs – These are the graphs BTER was designed to model well

2. Mesh graphs – These graphs correspond to meshes from physical problems (FEM or FD) and are very structured

3. Road network – These graphs came from the SNAP collection and are roads networks in U.S. states

4. Real-space pseudopotential method – These graphs come from linear systems that appear in solution to some physics problems

5. "Other" graphs – These are just a myriad of the other types of graphs on the Florida Sparse Matrix Archive

Generated graphs were created where possible from given graphs by following the steps outlined on the BTER website [5].

Graphs were compared to their generated versions in two ways: visualizing degree and clustering distributions (which were given as input to the algorithms) and visualizing comparisons of the eigenvalue spectra (which were not given as input).

If BTER was unable to generate the graph, no comparison was made and the graph was marked as such.

3 Results

This section contains the results of the experiments described above.

3.1 Initial Graph Generation Test

Number	Description	BTER worked
	Social Networks	
2297	Collaboration network of Arxiv General Relativity	Yes
980	Stanford-Berkeley web (Not symmetric)	Yes
	FEM	
33	Symmetric Stiffness Matrix, Ore Car (Lumped Masses)	Yes
2283	3D FEM, transient electric field diffusion. Evan Um, Stanford	Yes
1421	Circuit simulation problem, Ufuk Okuyucu, AMD, Inc.	No
1892	Denormals in $A + alpha * I$ as $alpha > 0$ varies, J. Castrillon, Teledyne	No
2258	FEM problem, temperature and deformation of a steel cylinder	No
430	Test Matrix from FIDAP: Ex33.mat	No
228	Splatzman Symmetric Finite Difference Three Ocean Model	No
29	Symmetric Stiffness Matrix, Medium Test, Consisten Masses	No
791	Acoustic Radiation around aft duct fan. D. Okunbor	No
	Road networks	
2317	Road network of California	Yes
2318	Road network of Texas	No
2319	Road network of Pennsylvania	No
	Psuedopotential Method	
1363	Real-space pseudopotential method. Zhou, Saad, Tiago, Chelikowsky	Yes
1366	Real-space pseudopotential method. Zhou, Saad, Tiago, Chelikowsky	Yes
1350	Real-space pseudopotential method. Zhou, Saad, Tiago, Chelikowsky	No
	Other	
1347	Sparse bundle adjustment, 3D vision, M. Lourakis, Greece	Yes
1551	IBM TJ Watson, non-linear optimization	Yes
4	Admittance Matrix 685 Bus Power System, D.J.Tylavsky, 1985	Yes
3	Admittance Matrix 662 Bus Power System, D.J.Tylavsky, 1985	Yes
2	Admittance Matrix 494 Bus Power System, D.J.Tylavsky, 1985	Yes

Both social networks worked, very few of the FEM models worked, and with other graphs, it was very hit and miss what worked (Only one of three road networks worked, but all power systems worked. Two out of three 'Real-space Psuedo...' matrices worked).

There were two common errors, which, according to correspondance with T. Kolda [3], both resulted because of the graphs' dissimilarities with social network graphs. First of all, the degree distributions of FEM graphs can be quite different to that in social networks. Displayed in Figure

r

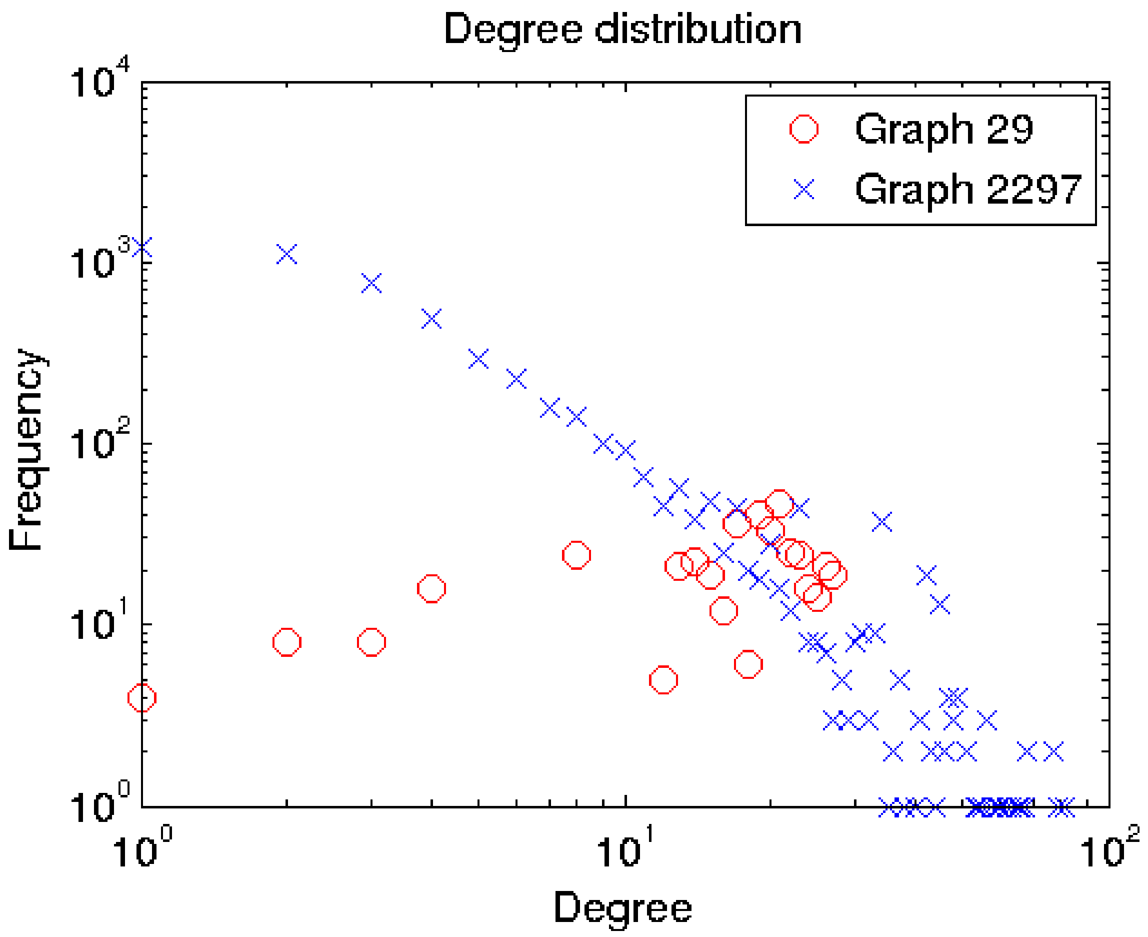


Figure 1: Comparison of the degree distributions of graph 29 (FEM) and graph 2297 (Citations)

1 is the degree distribution of graph 29 (FEM) alongside the degree distribution of graph 2297 (Arxiv collaboration network). BTER is designed to model the long decaying tail of graph 2297, not the slowly growing tail of graph 29. According to T. Kolda (correspondence), many of the errors in matching these graphs were due to the non-decaying degree distributions. Part of the BTER algorithm is a choice of probability distribution function that represents an edge probability for a cluster given the degree distribution of the original graph. Perhaps choosing this more carefully would overcome this limitation.

Secondly, FEM meshes can be unusually connected. The clustering coefficient is the ratio of triangles formed between edges with neighbors and then the number of possible triangles that could be formed by edges with neighbors. In FEM meshes, this is frequently 1. In social networks, this is frequency between 0.1-0.5 [3]. Because of how the BTER graph generate generates its subclusters, it does not allow fully connected clusters to be generated (from correspondence with T. Kolda, it might take a very long time to generate a fully connected cluster randomly).

Both of these errors seemed to be a result of how the BTER example code was implemented though, not the algorithm.

These errors popped up sporadically in other types of graphs. Very long range connections seemed to work better with the generator. This corresponds to filling rows with widely spaced elements, or adding edges that shortcut otherwise long paths through the graph. Even graphs that looked very irregular and generatable at face value, like the road maps, sometimes were not able to be duplicated with the BTER generator. Perhaps this could be pinned to the lack of long range graph connections.

3.2 Distributions, Smallest Eigenvalues, and Fiedler Eigenvectors

The degree and clustering coefficients, ten smallest eigenvalues, and Fiedler eigenvectors were computed for three graphs. The degree and sample coefficients were computed with code given in the BTER implementation [4], and the eigenvalues and eigenvectors were computed with code from Gilbert [2]. These are graph 2297 (Arxiv Citations), graph 33 (FEM), and graph 4 (Admittance matrix). The solvers frequently complained that the matrices were badly scaled or otherwise not easy to compute eigenvalues and eigenvectors for, but no effort was taken to make sure the solvers were working properly. In all cases, while the degree distributions and clustering coefficient graphs line up, the spectral measurements (either eigenvalues or eigenvectors) do not totally align. For graph 2297 (Arxiv Citations), the Fiedler eigenvalue is reasonably close (0.000673 compared with 0.000525), but as discussed later this is probably just a fluke.

Of note, the eigenvalues for the random graph are mostly below the eigenvalues for the original graphs. Also, for graph 4 and 33, the eigenvectors returned as the Fiedler eigenvector are very close to being the constant vector. This hints that maybe something is going wrong.

In some sense, the graph generator fits the distributions that it fits, and with some luck the spectral stuff can match up, but it is not guaranteed, and the match is not very pretty.

3.3 Fiedler Eigenvalues

Without requiring much effort, the Fiedler value and condition number can be compared for a wider array of graphs than those above. The graphs in the table below were all selected primarily due to their small sizes. The eigenvalue solver frequently complained about badly scaled matrices, but nothing was done to check if the eigenvalues were really accurate or anything.

r

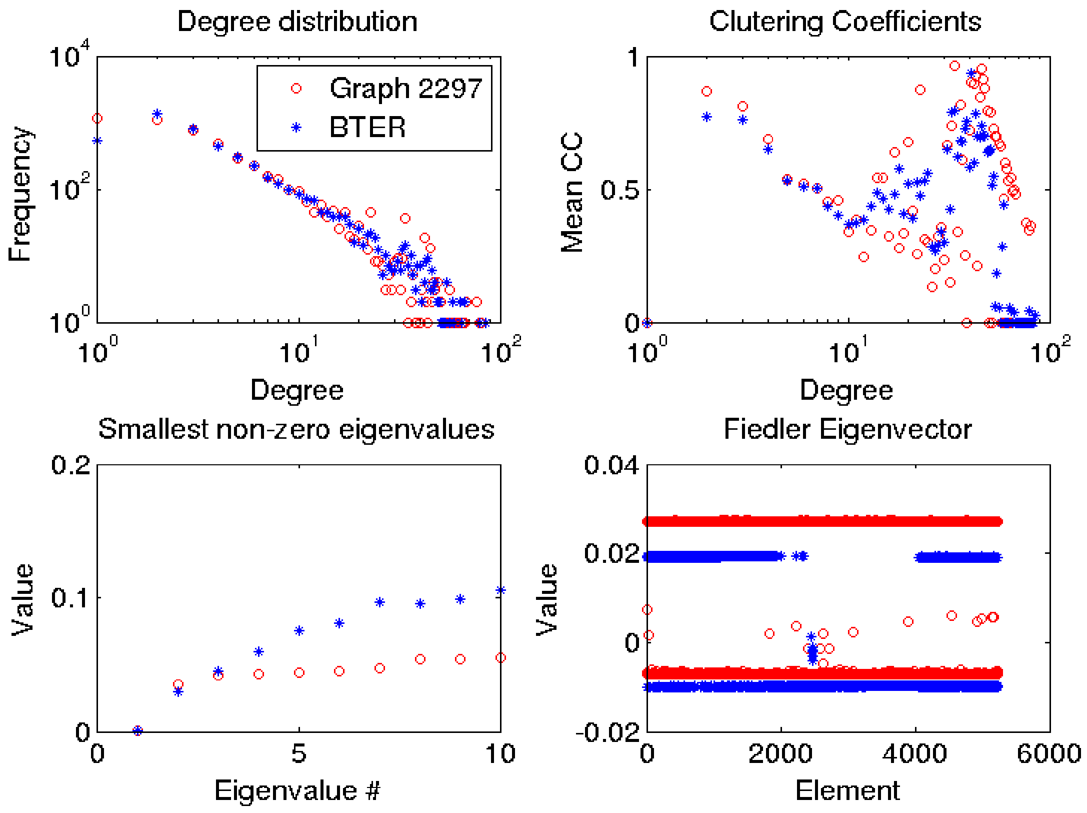


Figure 2: This is the comparison of generated and original graph 2297 (Arxiv Citations). The Fiedler eigenvalues are 0.000673 for the original graph and 0.000525 for the generated graph. The eigenvectors of the generated and original graph look similar except for a large chunk missing from the generated graph. It is unclear what causes this

r

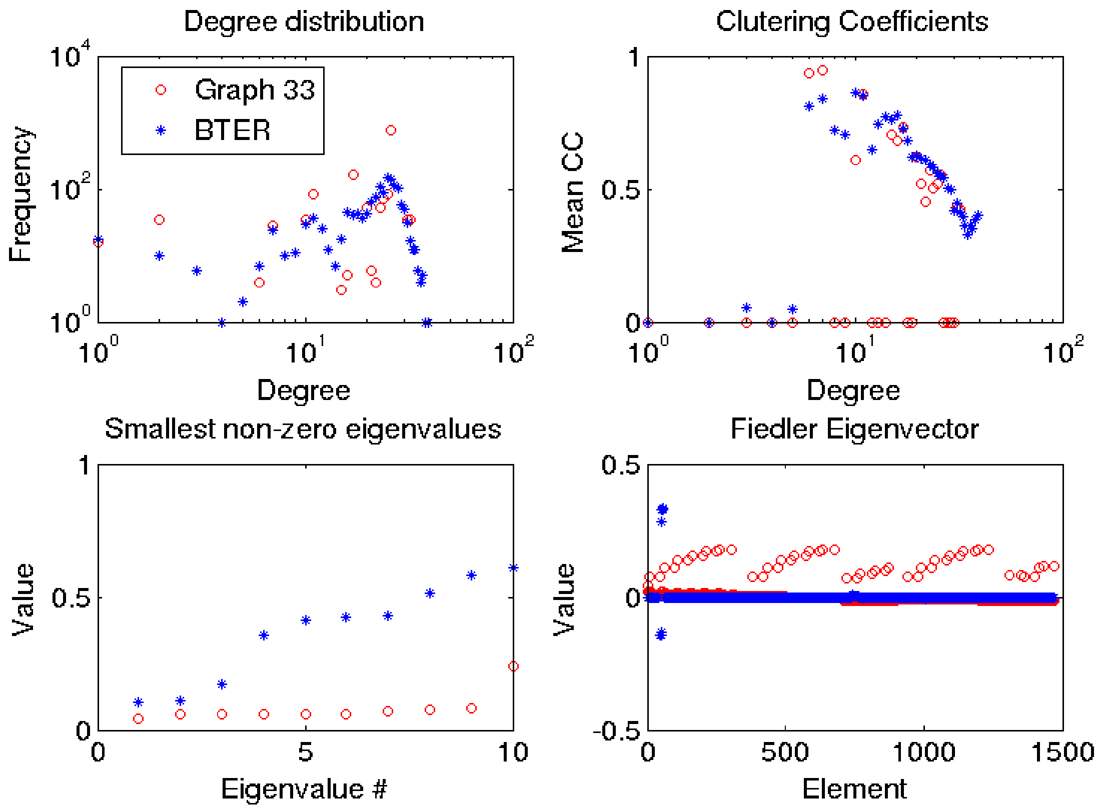


Figure 3: This is the comparison of generated and original graph 33 (FEM). The Fiedler eigenvalues are 0.041663 for the original graph and 0.106670 for the generated graph. For the FEM graph, a number of the degree vertices have clustering coefficients of zero. This phenomena showed up in many other systems, and can occur in a mesh built from squares, for instance. The BTER graph generator fits the degree distribution fine, even though the graph tail distribution does not look like a social network. Finally, the eigenvector for the real system has 'fins' which, depending on the run, pointed either up or down. This indicates that the eigenvector solver was struggling for these graphs (since mathematically there is only one Fiedler eigenvector). The generated graph did not duplicate this

r

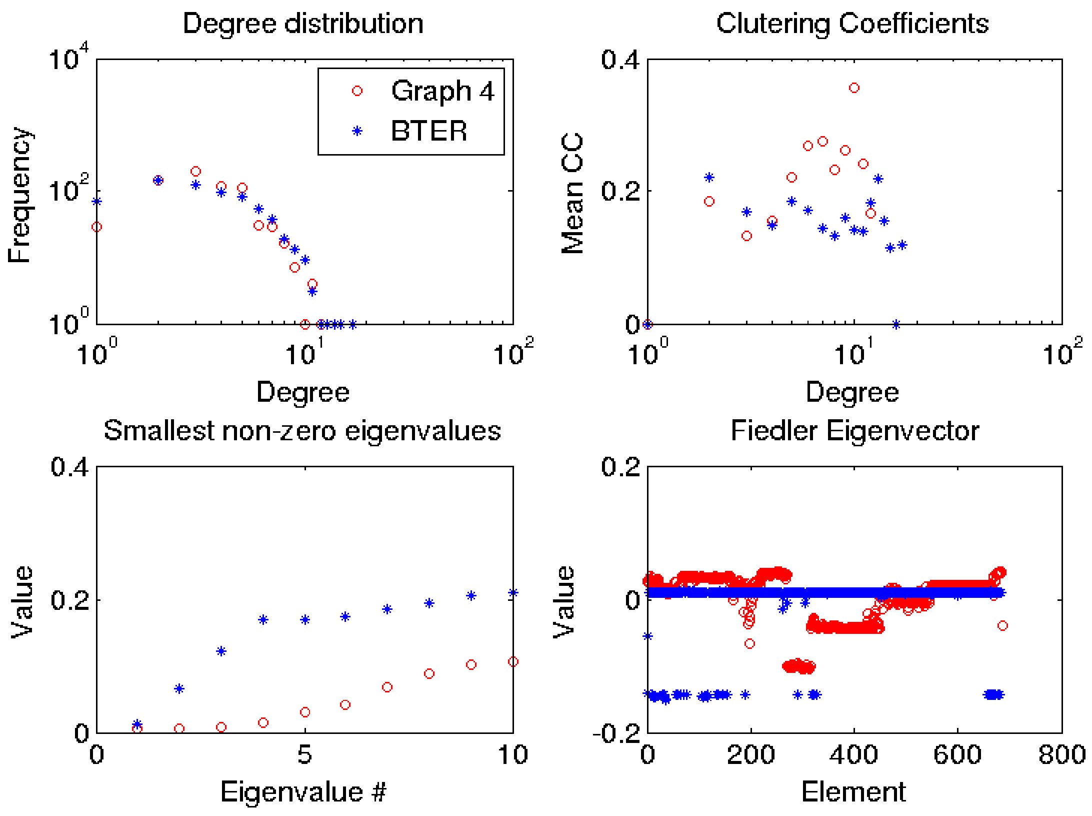


Figure 4: This is the comparison of generated and original graph 4 (Admittance matrix). The Fiedler eigenvalues are 0.006088 for the original graph and 0.013323 for the generated graph. The Fiedler eigenvector and all computed eigenvalues are quite different from the references

r

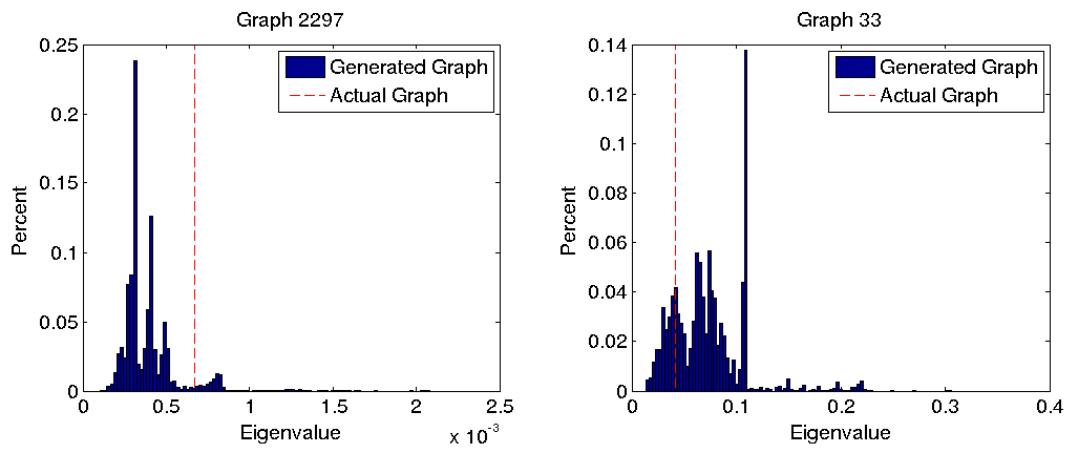


Figure 5: The Fiedler values from graph 2297 and graph 33 are compared to the ones from the generated graphs

Number	Real Fiedler	Generated Fiedler	Real Cond. Num.	Generated Cond. Num.
		Social Networks		
2297	0.5195706229	0.0039357127	3830.1330414811	281788.384360883
		FEM		
33	0.0416627326	0.0357234377	1630.1581377668	1296.5005986051
		Psuedopotential Method		
1366	2.4995347435	1.9632425369	165.8133466451	122.1822376171
		Other		
1347	1.5715035978	1.3704721957	1092.5841992422	596.4449019947
1551	0.5857864376	0.0023078737	3397.1845065909	464990.682703995
4	0.0053413717	0.009433869	3518.7665950482	1709.2210936338
3	0.013451767	0.0076989114	1087.129841142	1651.3912761102
2	0.0073634699	0.0061492206	1828.7851489422	2034.1680963568

This table shows mixed results. In some cases (graph 2) the Fiedler eigenvalues are quite close, but sometimes for similar graphs it is totally off (graph 3). The condition numbers tell the same story (being the radio of eigenvalues, this could be expected).

Naturally, the graphs generated here are random, and so the Fiedler values will have some varation. To capture this, the Fiedler eigenvalues of ten thousand generated copies of graphs 2297 and 33 were compared against their exact values. This is shown in Figure 5.

The first thing to note about the histograms is that the eigenvalue distributions are not obviously standard distributions, even given the relatively high resolution experiment (10000 samples & 100 bins). Secondly, picking an eigenvalue at random, there is very little reason to think that it will be very close to the correct eigenvalue.

r

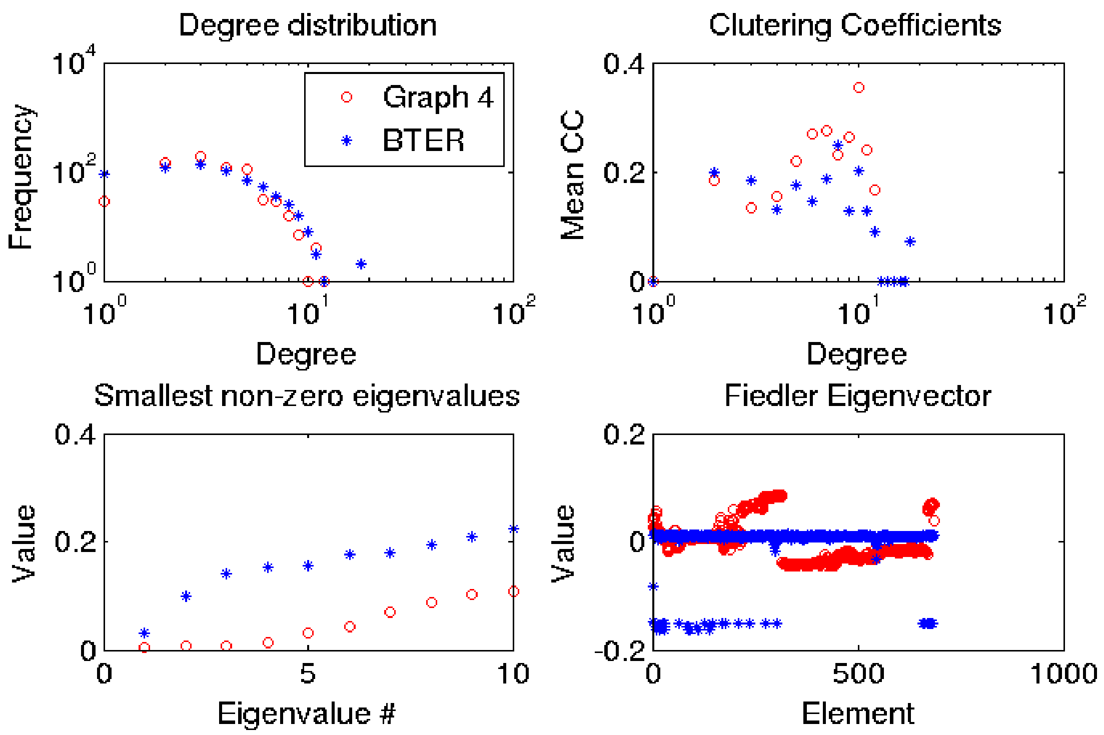


Figure 6: This is the equivalent to 3 (with Laplacian replaced by normalized Laplacian). This is the comparison of generated and original graph 33 (FEM). The Fiedler eigenvalues are 0.00534137 for the original graph and 0.0296625 for the generated graph. For the FEM graph, a number of the degree vertices have clustering coefficients of zero. This phenomena showed up in many other systems, and can occur in a mesh built from squares, for instance. The BTER graph generator fits the degree distribution fine, even though the graph tail distribution does not look like a social network. Finally, the eigenvector for the real system has 'fins' which, depending on the run, pointed either up or down. This indicates that the eigenvector solver was struggling for these graphs (since Mathematically there is only one Fiedler eigenvector). The generated graph did not duplicate this

3.4 Normalized Laplacian

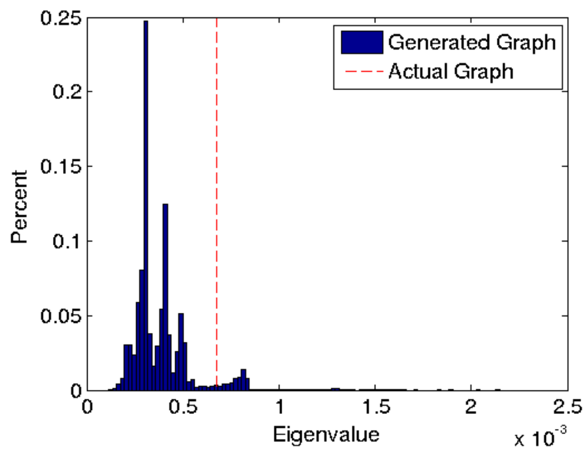
Throughout this discussion, the Laplacian matrix has been used for the spectral computations. Similar results are found for the Normalized Laplacian. Figure 6 is the equivalent of Figure 3, and Figure 7 is the equivalent of Figure 5 with the normalized Laplacian used for spectral comparisons.

4 Discussion & Conclusion

In conclusion, there was no dependable magic in the generated graphs. However, the spectral results were not terrible. If an error of a factor of two or three, then maybe the spectral results here are useful.

The BTER example implementation ran sucessfuly on about half of the graphs pulled from the Florida Matrix Archive. It did an admiral job of matching degree distributions and clustering

r



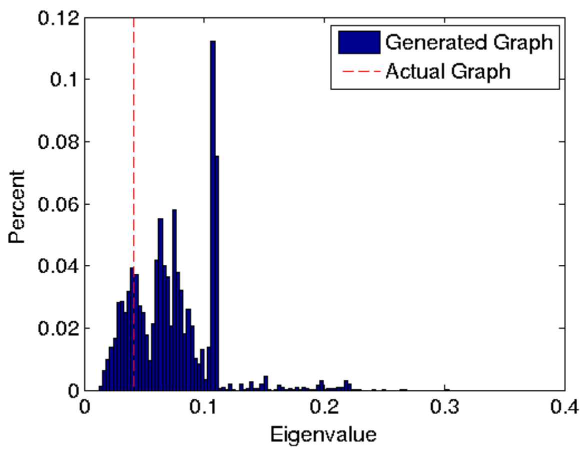


Figure 7: This is the equivalent to 5 (with Laplacian replaced by normalized Laplacian). The Fiedler values from normalized Laplacian of graph 2297 and normalized Laplacian of graph 33 are compared to the ones from the normalized Laplacians of the generated graphs

coefficients even in situations that it was not advertised to work in. If the issues with high clustering coefficients and non-decaying degree distributions could be resolved (and it seems like the limitations are simple implementation details – not algorithm roadblocks), it could be more useful.

References

[1] Timothy A. Davis and Yifan Hu. The University of Florida Sparse Matrix Collection. *ACM Transactions on Mathematical Software*, 38(1):1:1–1:25, November 2011.

[2] John Gilbert and Shanghua Ten. 'matlab matrix library'. `http://www.cs.ucsb.edu/~gilbert/cs290hSpr2014/Matlab/`. Copyright Xerox, John Gilbert, Shanghua Ten, Accessed: 2014-3-31.

[3] C. Seshadhri, Tamara G. Kolda, and Ali Pinar. Community structure and scale-free collections of erdos-rényi graphs. *Phys. Rev. E*, 85:056109, May 2012.

[4] Ali Pinar Tamara G. Kolda et al. Feastpack v1.1, sandia national laboratories. `http://www.sandia.gov/~tgkolda/feastpack/`, January 2014. SAND2013-4136W, Accessed: 2014-6-1.

[5] Todd Plantenga Tamara G. Kolda, Ali Pinar and C. Seshadhri. Bter guide: How to create a bter graph that matches an existing graph. `http://www.sandia.gov/~tgkolda/feastpack/doc_bter_match.html`, January 2014. SAND2013-4136W, Accessed: 2014-6-1.