Analysis of the Amazon Book Reviewer Community

G.A. Kohring

1 Obtaining the Data

The goal of this project is to illuminate the community structure in the world of the Amazon book reviewers. To that end, we use the raw data available from the SNAP (Stanford Network Analysis Project) data repository [1]. Among other datasets, SNAP contains a collection of product reviews scraped from the Amazon web site. The timespan over which the latest version of the book review dataset was collected covers June 1995 through March 213 and consists of 12,886,488 reviews of 859,772 books from 2,588,991 reviewers. A single 14 GB file holds the entire dataset arranged in a simple list format.

1.1 Cleaning the Data

The original dataset was created for a different purpose, hence it has has to be cleaned up for use in this project.

As the reviews are listed by the Amazon Standard Identification Number (ASIN) of the book they are reviewing, it helps to understand that Amazon assigns each edition of a book a different ASIN. Hardcover, paperback, and kindle editions of the same book all receive different ASINs. However, Amazon recognizes that these different editions are the same book and links them together so that the same reviews appear under each edition. Thus, in the SNAP raw data, a single review might appear multiple times in the list. To avoid skewing the results, such multiple reviews should be counted as a single review and the reviews for all editions of the same book consolidated under a single title.

Multiple copies of a single review also occur if the reviewer edits a previous review, e.g., to fix typos or add additional information. These updated reviews appear

in the list with a different time stamp. Again they need to be coalesced into a single review to avoid biasing the data.

Another problem is that some reviewers prefer to remain anonymous and are listed in the raw data as "unknown". As it is not possible to differentiate between the many anonymous reviewers, all anonymous reviews should be purged from the data set.

The cleaning algorithm is then:

1. Extract metadata for each review and place it into tabular form with one row per review and the following columns
   - ASIN
   - Title
   - Reviewer ID
   - Reviewer Screen Name

   (The actual review itself is not needed for this project.)
2. Delete Anonymous Reviews
3. Consolidate multiple reviews into a single review
   (a) For each ASIN make a list of all its reviewers.
   (b) For each title make a list of all books with the same title and different ASIN.
   (c) If two books with the same title but different ASINs have the same reviewers, treat them as the same book.

The consolidation step is complex because the raw data does not contain author information, otherwise we could simply merge review lists from all books with the same title and same author.

After cleaning, 8,777,342 reviews from 2,588,990 distinct reviewers remain in the table.

1.2 Creating the Network

To create the network, we assign each reviewer to a vertex and then make edges between vertices when the corresponding reviewers have both reviewed the same book. For every book the two have reviewed in the common, the weight is increased by one. Once completed, we are have a large, undirected network with the properties shown in Table 1.

Table 1: Global properties of the network.

vertices	2,588,990
edges	436,259,145
density	1.30×10^{-4}

1.3 Examining the Data

Before analyzing the network itself, it is interesting to look at a few statistics from the data. Table 2 lists the top ten reviewers in terms of the number of books they reviewed. We will return to this data in the next section when we look at network methods for ranking reviewers.

There is a wide variation in the number of books reviewed by each reviewer, with the mean being 3.39. The median number of reviews per reviewer is 1. Just 5.4% of the reviewers account for 50% of all reviews.

In addition to individual reviewers, the list contains groups like the *Midwest Book Review*, a non-profit organization dedicated to promoting literacy [2]. One is tempted to purge all group accounts from the network, however, without extensive background checks on each reviewer it is difficult to separate groups from individuals.

Table 2: Rank by Number of Reviews.

Rank	Screen Name	Number of Reviews
1	Midwest Book Review	21,695
2	Harriet Klausner	13,781
3	Shalom Freedman	6,390
4	Blue Tyson	6,067
5	Donald Mitchell	4,945
6	Charles Ashbacher	4,646
7	Steven H. Propp	3,534
8	John Matlock	3,312
9	S. Schwartz	3,279
10	E. A Solinas	3,085

2 Analysis

This is a large network and since the complete graph does not fit in the RAM on the machines at my disposal, tools like *Gephi* cannot be used. Instead, hand written C++ code is required to handle the analysis.

2.1 Degree Distribution and Degree Centrality

The degree distribution can be calculated by summing up the number of edges attached to each vertex and doing so leads to the diagram depicted in Fig. 1. In class we learned that the degree distribution often has a heavy tailed that can be described by a power law. The red line in the fig. 1 is a Maximum Likelihood Estimate (MLE) of a power law fit to the data:

$$p(d) = \frac{\alpha - 1}{d_{\min}} \left(\frac{d}{d_{\min}} \right)^{-\alpha}, \tag{1}$$

where $\alpha = 2.86$ and $d_{\min} = 6153$. $d_{\min}$ was obtained by varying $d_{\min}$ in eq. 1 and selecting the value of $d_{\min}$ that minimizes the Kolmogorov-Smirnov statistic. While there is a good fit over a few decades in the middle of the distribution,

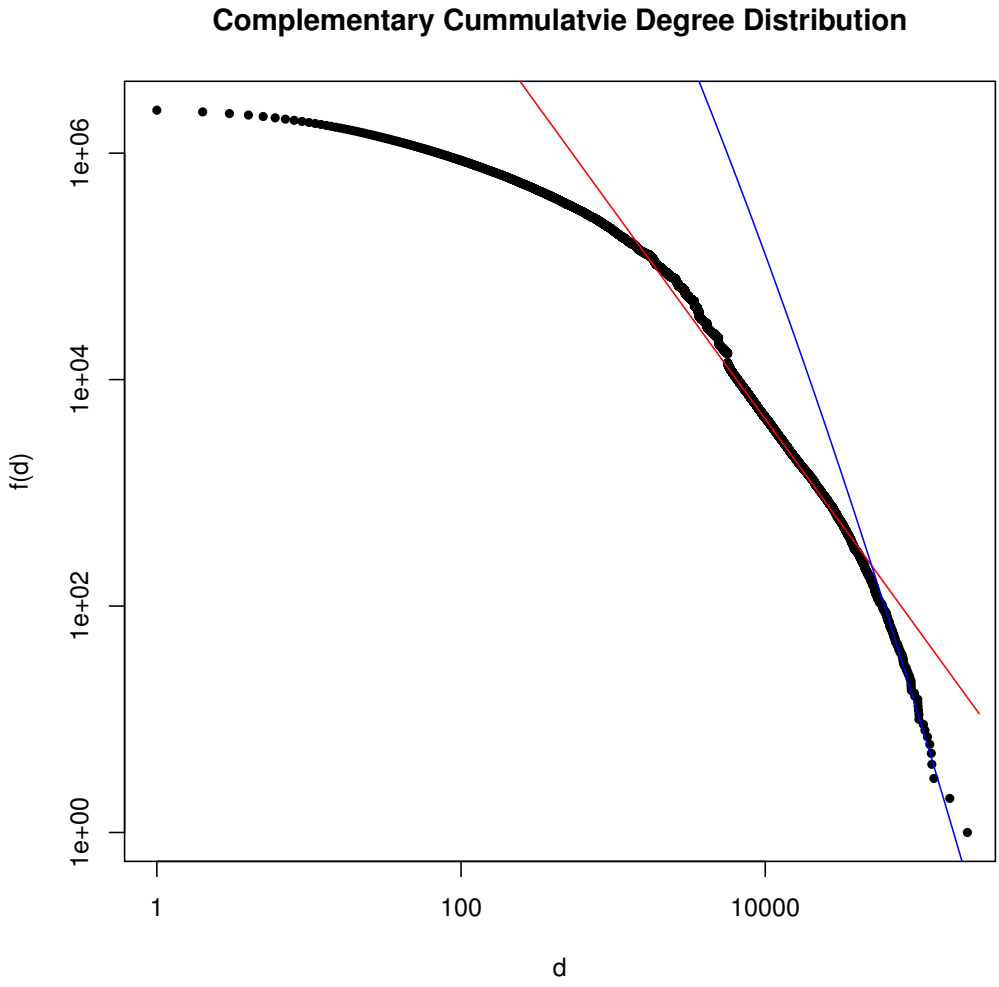

Figure 1: Vertex degree distribution. The red line is an MLE fit to a power law and the blue line is a regression fit to a log-normal.

the tail falls off faster than anticipated and is better matched with a log-normal distribution as seen by the blue line in Fig. 1.

Degree centrality measures a vertex's importance based up its degree. In this network, degrees are in the range $[0, 259470]$. According to this criterion, the most important vertex in the network would the one with the maximum degree. The top ten vertices based upon degree centrality are shown in Table 3

Table 3: Rank by Degree Centrality.

Rank	Screen Name
1	Midwest Book Review
2	Harriet Klausner
3	Donald Mitchell
4	Brett Benner
5	Bookreporter
6	Busy Mom
7	CoffeeGurl
8	Robert P. Beveridge
9	Blue Tyson
10	Notnadia

Compared the ranking in this table with that of Table 2. While the top two positions are the same, only four of the previous top ten remain. A high rank in terms of the number of reviews, but a low rank in terms of degree centrality indicates the reviewer is reviewing many books other people seldom read. A high degree centrality but a smaller number of reviews indicates a reviewer who is reviewing the most popular books.

2.2 Eigenvector Centrality

As discussed in class, the eigenvector centrality measure ranks vertices according to the magnitude of their corresponding component in the principal eigenvector. The principal eigenvector can be efficiently calculated even for large matrices using the power iteration algorithm [3]. The top ten vertices based upon eigenvector centrality are shown in Table 4

Table 4: Rank by Eigenvector Centrality.

Rank	Screen Name
1	Blue Tyson
2	E. A Solinas
3	bernie
4	CoffeeGurl
5	Dave_42
6	M "CultOfStrawberry"
7	Ash Ryan
8	J. Harrison
9	Notnadia
10	Sean K

Note that only two of those ranked in the top ten according to degree centrality are still ranked in the top ten according to eigenvector centrality. And *Blue Tyson* is the only reviewer listed in the top ten on all three tables.

2.3 Cohesion

The reviewer network is not fully connected. Some statistics characterizing the network's cohesion are given in in Table 5. As we can see by the average cluster size, most of the clusters involve only a few individuals while the giant cluster encompasses 95% of the network.

The fragmentation value measures the proportion of pairs of vertices unreachable from each other. Fragmentation ranges from 0 (all vertices reachable) to 1 (all vertices are isolated). For an undirected network, the fragmentation can be calculated as:

$$F = 1 - \frac{\sum_{k=1}^{N_c} S_k(S_k - 1)}{|V|(|V| - 1)}, \tag{2}$$

where, S_k is the size of the k-th cluster and N_c is the number of clusters. For an Erdös-Renyi random network, we would expect $F \to 0$ when $|E| > |V| \ln(|V|)$.

Table 5: Cohesive properties.

number of clusters (> 2 vertices)	33,152
average cluster size	75.5
median cluster size	2
vertices in giant cluster	2,406,908
Fragmentation	0.136

2.4 Communities

Community detection is an ongoing research topic. The *fast-greedy* algorithm used in the *igraph* package has computational complexity $O(|E| \ln^2(|V|))$ and is too slow for large networks. *Gephi* uses the *Louvain* method [5] which appears to be faster than *fast-greedy*, however no complete analysis of its computational costs exist. For very large graphs, *label-propagation* [4], which was not mentioned in the lectures or homework, is the best alternative despite its short comings because it is simple to implement and requires only $O(|E|)$ operations.

The results of applying *label-propagation* are shown in Table 6. As can be seen, *label-propagation* is unable to find a more complete disentanglement of the largest community. This is a well known limitation of this method. There are several proposals in the literature to get around this limitation, but they go beyond the scope of this project. Apart from the largest community, *label-propagation* does a good job of recognizing community structure as evidenced by the relatively large modularity value for nodes outside of the largest community.

Table 6: Community properties.

number of communities	363,128
average community size	7.13
vertices in largest community	1,091,553
modularity	0.023
modularity (omitting largest)	0.755

Leaving the largest and smallest communities aside for the moment, we take a closer look at intermediate size communities, i.e., the set of communities, $\{c_i : |c_i| \in [100, 1000]\}$. A visualization of these intermediate size communities created with the OpenOrd software [6] is shown in fig. 2.

In this figure the separate communities and the direct links between them are clearly visible. In the complete graph, the communities are not disconnected, they are joined by longer paths through intermediate vertices in the main cluster which, for purposes of clarity, is not shown.

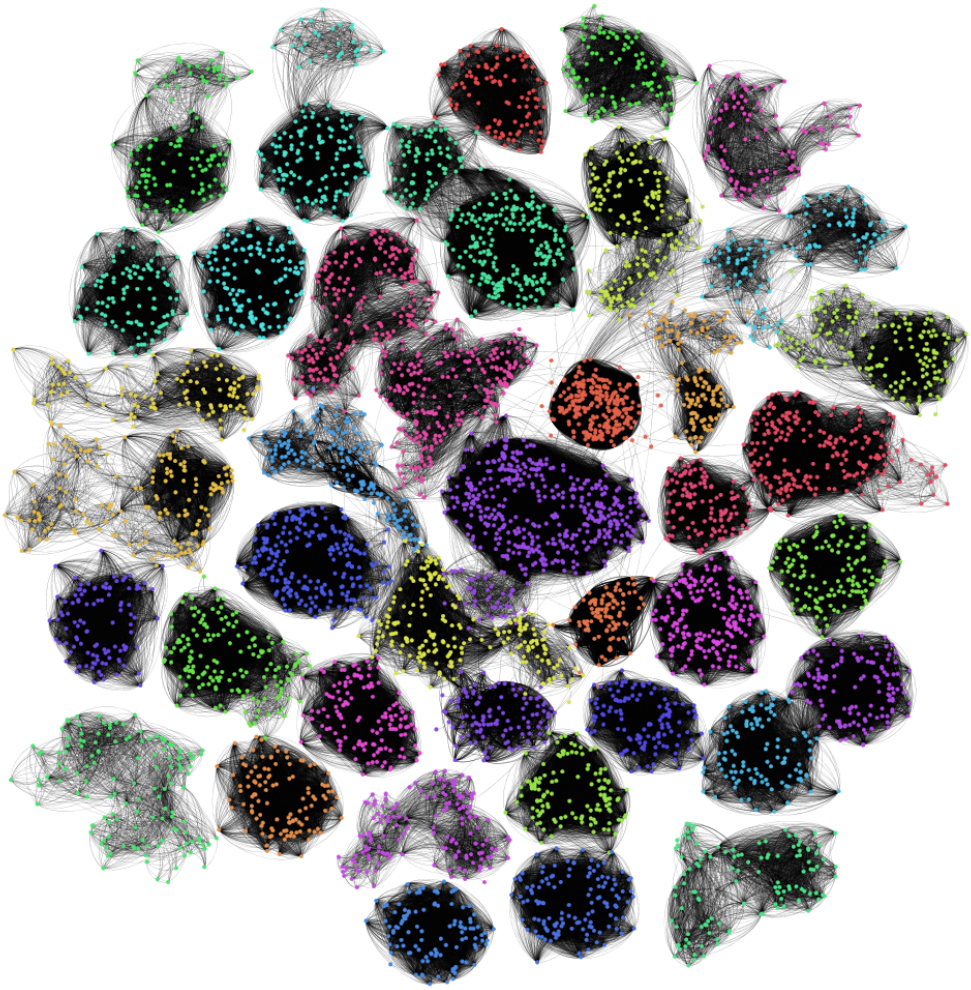

Figure 2: A sample of reviewer communities with $\{c_i : |c_i| \in [100, 1000]\}$. The colors correspond to community labels and the black lines represent links between vertices.

3 Discussion

The foregoing analysis has illuminated the community structure in the world of the Amazon Book reviewers. We have seen that those reviewers who review the most books are not necessarily the highest ranked when alternative, network based centrality measures are used.

The picture in fig. 2 shows how a well defined community structure has emerged in the reviewer network. In doing so, it offers an alternative method to make book buying recommendations. Instead of recommending books similar to what a reviewer has recently read, recommendations can be made based upon what people in the reviewer's community are reading.

Further analysis is possible. It would bee interesting to match the communities found in the previous section to Amazons's book subject categories to see whether people tend to review the same types of books.

The raw data contains the ratings the reviewers gave a book on a scale of 1-5. This information was not incorporated into the current analysis. It could be used for example to adjust the edge weights, or connect reviewers only if they both liked the same book.

Amazon also allows people to provide feedback on the reviews in terms of clicking a box indicating whether they found the review helpful or not. This information is available in the raw data, but was not incorporated into the current network. This information could be used to create a network of only those reviewers who generate high quality reviews.

As mentioned above, the software used to analyze the graph was written in C++ to handle the out-of-core storage. The interested reader can examine it on the github repository https://github.com/gkohri/azrev. It is more than 1,000 lines of code.

References

[1] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, 2014.

[2] J. A. Cox, "The midwest book review," http://www.midwestbookreview.com/, 2014.

[3] Wikipedia, "Power iteration — wikipedia, the free encyclopedia," 2014, [Online; accessed 13-November-2014]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Power_iteration&oldid=633515118

[4] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E*, vol. 76, p. 036106, Sep 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.76.036106

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008. [Online]. Available: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008

[6] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack, "Openord: an open-source toolbox for large graph layout," *Proc. SPIE*, vol. 7868, pp. 786 806–786 806–11, 2011. [Online]. Available: http://dx.doi.org/10.1117/12.871402