Using Amazon Product Review Models to Characterize Amazon Reviewer Communities

Alejandro Ceballos, Michael Chang, Justin Lee (Group 40)

December 9, 2014

Abstract

More often than not, the time component of Amazon reviews is overlooked for the actual content or the characteristics of the reviewer. This paper examines the relationship between timestamp of reviews, reviewers, and products in a large dataset of Amazon reviews. Our model slices the graph into years, so that we can explore the development of Amazon user communities between 1996 and 2005. We create our graphs by generating edges between users who reviewed the same items, and we use review metadata to filter our edges in order to create models of helpful, positive, and negative reviewers. By applying community detection to these graphs, we explore the way these different communities grew and changed over time. We found communities around different product categories that changed over the years according to which items were more popular at the time.

Introduction

0.1. Motivation

This project aims to better understand Amazon reviewer communities using graph models that represent Amazon reviews. Previous studies have explored the implications of review helpfulness (as rated by users) or have conducted sentiment analysis on review content to classify a particular review. Other studies have heavily focused on better understanding product communities, for applications such as improving recommendation engines.

Our approach attempts to describe and better understand a much more abstract entity: the Amazon reviewer. What can we learn about Amazon reviewers based off of what products were reviewed and how highly rated these products were? Can we identify distinctly similar reviewers based off of their attitudes towards the products that were reviewed in common? Perhaps if we can define communities between reviewers based off of the shared product group, we might be able to suggest deeply similar social friends according to the community attitudes towards certain products.

Moreover, we aim to better understand how these descriptions of Amazon reviewer communities change over time. For instance, networks based off of product reviews may reveal a correlation between the users who review the same product, and especially within the same time period. A network representing product reviews of the same product during the same time slice effectively resembles a similarity graph, allowing detection of user communities with particular characteristics. We anticipate that our characterization of communities will yield significantly varied results based off of the year as well as the time period within the year.

Analysis of how these network characteristics change over time may offer insights into the strength of a particular reviewer community, the composition of communities overall, in addition to how the attitudes towards certain products might change within and across the years.

0.2. Problem Definition

Ultimately, we came to a single question that united our line of reasoning and on which our research will focus on: *How might we characterize the communities of Amazon users under different views/models of the Amazon product reviews?*

This question requires that we properly define the relevant pieces of our problem statement. We define “characterize” as a mix of quantitative and qualitative metrics that help to visualize the user nodes for a particular graph model. Quantitative metrics include metrics such as graph diameter, PageRank, and SalesRank, which is a measure of an Amazon product’s popularity. We define “model” as a graph of Amazon reviewers with edges that connect two users when both users satisfy a certain predicate (e.g. both users ranked the same product highly, poorly, or the same). We define “community” as groups of Amazon reviewers that are defined according to a particular algorithmic approach (e.g. using a baseline or a most sophisticated community detection algorithm). Thus, the model chosen will impact the communities that are subsequently characterized by the community detection algorithm.

Prior Work

Below is our analysis on three papers that have benchmarked previous network analysis on various forms of the Amazon graph.

1.1. How opinions are received by online communities: A case study on Amazon.com helpfulness votes [2]

Summary

This paper explores some factors that may affect the perceived helpfulness of book reviews on Amazon. They start by introducing a number of hypotheses from sociology and social psychology that may explain which reviews users will find helpful. One hypothesis is that users will find reviews more helpful if the reviews agrees with other reviews for the product (conformity). Another is that more negative reviews will be judged more helpful (brilliant-but-cruel), and one final hypothesis is that users will think a review is helpful if they agree with the reviewer's rating of the product (individual-bias). Of course, the authors must control for the possibility that perceived helpfulness can be predicted directly from the text of the review--that some reviews are objectively considered "more helpful" due to what they say.

Because many reviews on Amazon have similar wording, the authors find that text alone does not explain how helpful users perceive a review. Instead, they argue that their findings support the individual-bias hypothesis: users who like a book are more likely to find positive reviews helpful, and users who do not like the book are more likely to find negative reviews helpful. An interesting aspect of their data is that the reviews users found helpful depended on the variance of the ratings of the book. When variance was low, reviews with the average rating tended to be seen as more helpful. On the other hand, when the variance of ratings was high, suggesting a very controversial book, users found reviews helpful if they tended to express a more positive or more negative opinion than the average.

Discussion

Overall, the paper seems to do a good job controlling for and testing various confounding factors that could explain their observations. Dividing the products up by variance seems to be a very useful insight that helps them present a unifying theory. Controlling for text using similarity ("plagiarism") allows them to show that opinions on helpfulness will differ on almost-identical reviews, suggesting that there must be another factor besides the text alone.

On the other hand, it isn't clear that nearly-identical reviews are the best representatives for review quality. For example, intuitively, one would expect that "brilliant but cruel" reviews would say something particularly insightful or original, which isn't necessarily replicated in other reviews. It would have been interesting to see the distribution of helpfulness for reviews that weren't plagiarized.

1.2. Finding community structure in very large networks [1]

Summary

The paper first focuses on the implementation of a new community finding algorithm, and then an example of if its use in a large Amazon items dataset. The algorithm runs in O(*nmdlogn*) time, where *n* is the number of vertices, *m* is the number of edges, and d is the depth of the dendrograph representing the communities. For many real-world networks, where *m ~ n* and *d ~ logn*, the algorithm runs in O($nlog^2n$). The algorithm categorizes communities using the modularity of the nodes. The Amazon dataset uses items as nodes and to connect the nodes uses undirected edges between an item *i* and the top 10 items purchased by users who also bought that item. The top 10 communities that are found consist of 87% of the vertices in the graph

Discussion

The paper relates to information that we've covered in class in that it focuses on how to find communities in networks. Finding and classifying communities is a good way to classify nodes into groups. Once nodes have been categorized, insights can be made about groups using metadata and how different groups act and interact.

This paper is strong in that it describes the implementation of its algorithm in great detail, and then uses a very relevant dataset to test the algorithm, however it weak in that it does not focus on the model for the algorithm. The paper faces is that it does not do a good job of addressing the characteristics of of the communities that it generates using modularity. The paper compares its algorithm's runtime to other classification algorithms' run times, but it does not discuss its community finding scheme to that of others algorithm's. The algorithm uses modularity to find its communities, but does mention whether modularity is as accurate as other schemes or if there is some sort of accuracy to runtime tradeoff being made. Another weakness that this paper faces is that it does not classify Amazon's dataset besides mentioning how many edges and vertices it has. The paper could of classified whether it ran in O(*nmdlogn*) time or O($nlog^2n$) time.

1.3. Trust and Helpfulness in Amazon Reviews: Final Report [3]

Summary

The authors attempt to gauge the trustworthiness of product reviews based on a combination of factors: the trustiness of a user, the honesty of a review, and the reliability of a product. The authors implemented a graph model called a *review graph structure*, which was then used to iteratively calculate the trustiness of a particular review using a custom algorithm until convergence was achieved. The subsequent trustiness scores of a particular review were used to distinguish spammers from legitimate reviewers.

Discussion

One of the limitations of the paper is that the data does not inherently provide any means to detect actual spam reviews in order to verify the accuracy of the proposed algorithm. This makes it difficult to evaluate whether the user trustiness values are correct, where correctness assigns high trustiness to legitimate users and low trustiness to spammers. As a result, the authors used a simulation to determine the algorithm's effectiveness by injecting fake users with fake reviews into their review graph structure. This strategy is somewhat unreliable because they randomly generated which products their dummy users would review; in reality, reviewers don't randomly choose reviews to rate and spammers might employ various strategies when determining which products to attack. A nice consideration was that the authors selected 4 different models of review spammers to mimic. However, there is not much consideration for more nuanced spamming behavior. For instance, none of the spammer models accounts for a spammer who might provide a rating different from the average rating and provide a rating that strongly deviates from the mean, but that is not necessarily a 1/5 or 5/5.

1.4. Further discussion

The papers focused on how users behave or interact with each other on Amazon, where the key interactions are centered around products. A few of the papers explored the use of novel algorithms to detect and gauge interesting phenomena such as user trust and review helpfulness.

Paper 1 and Paper 3 both take different perspectives on evaluating the trustworthiness of reviews. Paper 1 and Paper 2 focus on the communities around items and reviews and the impact of these communities on the reviews. Finally, we see that Paper 2 and Paper 3 leverage interesting algorithms to detect communities with reviewers and additionally provide a heuristic for a user's trustworthiness; this in turn encourages us to think about the importance of efficiency when developing our own algorithms for analysis.

Paper 2's is explicitly relevant to the topic of detecting communities, a useful technique for analyzing communities. Moreover, it discusses useful/fast algorithms for large-scale networks. Paper 3 demonstrates construction of a novel indirect tripartite graph, with nodes representing users, reviews, and products.

Methods

2.1. The Data

The dataset we use is the Amazon purchasing metadata,[1] which consists of data for 548,552 products (books, DVDs, music CDs, and videos) and 7,781,990 reviews on these products. The data includes the salesrank of each product and a detailed categorization. It also contains similar products based on co-purchases. Reviews have date information and can be associated to their reviewer by unique user ID. Each review also has the number of positive and total helpfulness votes. We do not have the text of each review.

We utilized a prepared dataset of Amazon purchased provided by 224W. This data was acquired by crawling the Amazon website in 2006 and includes reviews from 1995 to 2005. We wrote a custom parser in order to extract out the metadata into a structured format for use in graph algorithms.

2.2. Mode

There are a couple of ways we considered modeling the dataset. One model considers products and reviewers as nodes in a bipartite graph, where we have a directed edge from a reviewer to a product he or she reviewed. This model would allow us to enumerate each product reviews in chronological order and have fine-tuned control over time slices; we could easily limit our search to earlier or later reviews of a particular item by only considering the first *k* edges (ordered chronologically) going into an item. This candidate model allows us to identify products with many reviewers in common or vice versa, which may be relevant when controlling for factors such as average ratings.

[1] http://snap.stanford.edu/data/amazon-meta.html

We ultimately chose to use a model that restricts nodes to only reviewers. This approach would allow us to characterize and better define Amazon reviewer communities in the graph by allowing more direct connections to exist between reviewers and within reviewer communities. This model also allows us to potentially represent sets of products related by certain external events (e.g. products that became popular at the same time). We use predefined time periods to filter our data into subgraphs of the larger network, allowing us to efficiently consider a wide range of various time periods for comparative analysis of community categorization. This model requires a systematic approach to edge generation, which we control via use of filters.

Filters

Our approach for generating our reviewer graphs is based on creating edges between user pairs that reviewed the same item(s). This model allows us to to generate a network that focuses on the relationship between users based off of a particular common perception towards an Amazon product. In order to explore the relationships between users and generate different interesting graphs, we filter the edge creation with different conditions.

The first and most overarching filter that we apply to our models is that of time. We use the time data of reviews that is part of our dataset in order to generate graphs based on the year in which the reviews were made. By doing this, we are able to split our data up into discrete chunks and we are able to see trends in Amazon's user communities over time. This information is relevant to us because Amazon's user base and items has been developing over time, which means we could see different trends amongst communities and items over time.

Besides the overarching way we time-slice our data by years, we use review metadata to add an extra layer of filtering to our graphs. We apply three different filters to our graphs, which we label as *helpful*, *negative*, and *positive*.

- The *helpful* filter is generated by only creating edges between users who reviewed the same item if their reviews were found to be helpful by more than 60% of users. We use this filter gain an insight about communities of helpful users
- The *positive* filter is generated by only creating edges between users who reviewed the same item if both reviewers left reviews of four or five stars. We use this filter gain an insight about communities of users who are very positive towards the same products
- The *negative* filter is generated by only creating edges between users who reviewed the same item if both reviewers left reviews of one or two stars. We use this filter gain an insight about communities of users that are very negative towards the same products

By using these filters we look at the way that specific groups of users behave and how their behavior changes over time. We are also able to compare these filtered communities to each other and see if these different types of user behave in different ways.

Difficulties

Some general difficulties with our problem include the massive graphs resulting from the parsing the metadata. Reading in such a large file required multiple processes for graph generation and data transformation.. In general, our runtime in python is too slow given the tremendous size of our graphs. This means that many community detection algorithms might not complete in a reasonable amount of time, so we had to find more specialized algorithms.

2.3. Algorithms

Community Detection

Immediate Neighbors

Using immediate neighbors as a community is a simple approach to trying to find communities with a graph. This approach is useful in that it is very simple to calculate a node's community, but it does not provide enough information about general communities at a higher level.

SCC

A strongly connected component (SCC) is a set of nodes for which every node is reachable by any of the other nodes. The maximum SCC is is the largest such set in the graph. We would expect that as the strongly connected component would be less sensitive to node deletion if the network has a normal distribution of connected nodes, since the attack policy removes highly connected nodes first.

```
MaxCCSize(V, E):
  nodesLeft := V
  maxCC := 0
  while nodesLeft is not empty:
```

```
            startNode := arbitrary node from nodesLeft
            run BFS on (V, E) starting from startNode
            reached := set of nodes reachable by BFS
            nodesLeft := nodesLeft \ reached
            if |reached| > maxCC: maxCC = |reached|
        return maxCC
```

Figure 1: code for finding the Maximum Size SCC

Girvan-Newman Algorithm

The Girvan-Newman community detection algorithm is based on finding the betweenness of edges, and removing the ones with the highest scores. While this method provides accurate results it has a relatively long running time. With a runtime of O(m^2n) on a network of n nodes and m edges, the algorithm proved intractable for our large dataset.

Infomap

The Infomap library [4] is closely based on the Louvain method of finding communities. The library implements a greedy algorithm which is based on maximizing the the modularity of communities. The modularity of a community is the density of edges within it versus the density of the edges between communities. By maximizing this modularity over all of the communities, we can achieve close clusterings of nodes that are related to each other.

All the nodes in the graph begin as separate modules. These modules are then randomly combined one by one with neighboring modules, slowly building up an optimal group of modules. At each step of the algorithm, the randomly picked module is combined with the module that would maximize the modularity of the graph

Equation 1: The algorithm operates under the following formulas:

$$\begin{aligned}
&\textbf{Modularity} : B_{ij} = A_{ij} - \tfrac{k_i k_j}{2m} \\
&\textbf{Membership} : s = \{-1, +1\} \\
&\textbf{Given} : Q(G, s) = \tfrac{1}{4m} \textstyle\sum_{i \in N} \sum_{j \in N} (A_{ij} - \tfrac{k_i k_j}{2m}) s_i s_j = \tfrac{1}{4m} s^T B s \\
&\textbf{Optimization} : \text{Find } s \in \{-1, +1\}^n \text{ that maximizes } Q(G, s)
\end{aligned}$$

Part of the Infomap algorithm is predicting how many levels of communities exist in the graph. The library can either estimate the optimal number of levels, or it can be fixed to two levels. Since we were looking for relatively small communities, and we did not want to examine larger overarching ones, we fixed the number of levels to two. This allowed us to find many small communities per graph, with a couple of larger more prevalent ones.

Equation 2: The algorithm uses PageRank to sort the communities that are generated

$$\textbf{PageRank} : r_j = \textstyle\sum_{i \to j} \beta \tfrac{r_i}{d_i} + (1 - \beta) \tfrac{1}{n}$$

The Infomap algorithm seemingly runs in O(nlogn), where n is the number of nodes in the graph, so it is able to give results for graphs of very large proportions.

Results and Findings

Comparison of Community Algorithms:

Table 1: Baseline vs Infomap community algorithms and their impact on community size in 2000.

	2000 q1	**2000 q2**	**2000 q3**	**2000 q4**

Baseline SCC	0.679	0.7223	0.7214	0.6737
Baseline Mean Degree Distribution	5	5	5	5
Louvain (level 2) Mean Community Size	31.295	70.92106	68.77874	68.45062

Consider a baseline algorithm for determining a reviewer's community. Simply considering a particular reviewer's neighbors seems incomplete because this definition of community is restrictive and not transitive. If we were to take each node's neighbors as a neighborhood, we would essentially be creating a single neighborhood for each node. While this would give us information about a node's neighbors, it would be difficult to commit a node to a single neighborhood, or get an accurate understanding of the nodes that are similar to a single node. We observed empirically in 2000 that there were a significant number of extremely small communities, with the mean degree distribution around 5 nodes or so (Table 1); communities of exceedingly small size have the tendency of potentially describing the community too specifically.

Looking at SCCs does not work as a good way to analyze communities because the graphs that we generate are so connected enough that the largest SCC is very dominant. For almost all years, we saw an SCC composed of over 80% of the total nodes (Table 1). Such a large SCC is created because we have a power-law esque degree distribution, and the fact that there are a few users who review a wide variety of different items. Categorizing these nodes as a community did not make sense, since this SCC captures the fact that there are a few people who like many different items, therefore linking everyone together and resulting in a diluted group of reviewers. Information about small communities would get lost in the large SCCs, so this approach proved to be suboptimal for us.

Infomap, proved to be most valuable for our study because it captured fine grain information of looking at a nodes direct neighbors while at the same time looking at connected components. The dendrographs generated by the Infomap algorithm provides a macro look of what the communities look like, similar to what the largest SCC would have given us. At the same time, cutting the dendrograph towards the leaf nodes provides a good picture of a node's close neighbors and community. This flexibility allowed us to tune the way we used our community detection algorithm to better characterize the community of Amazon reviewers.

Because we did not have any information about reviewers themselves, we characterized communities by their **representative item set**. An item was considered representative of a community if two reviewers in the community both reviewed that item satisfying the predicate of our model. Using this, we were able to analyze the quality of the communities we found using both quantitative and qualitative metrics on these item sets.

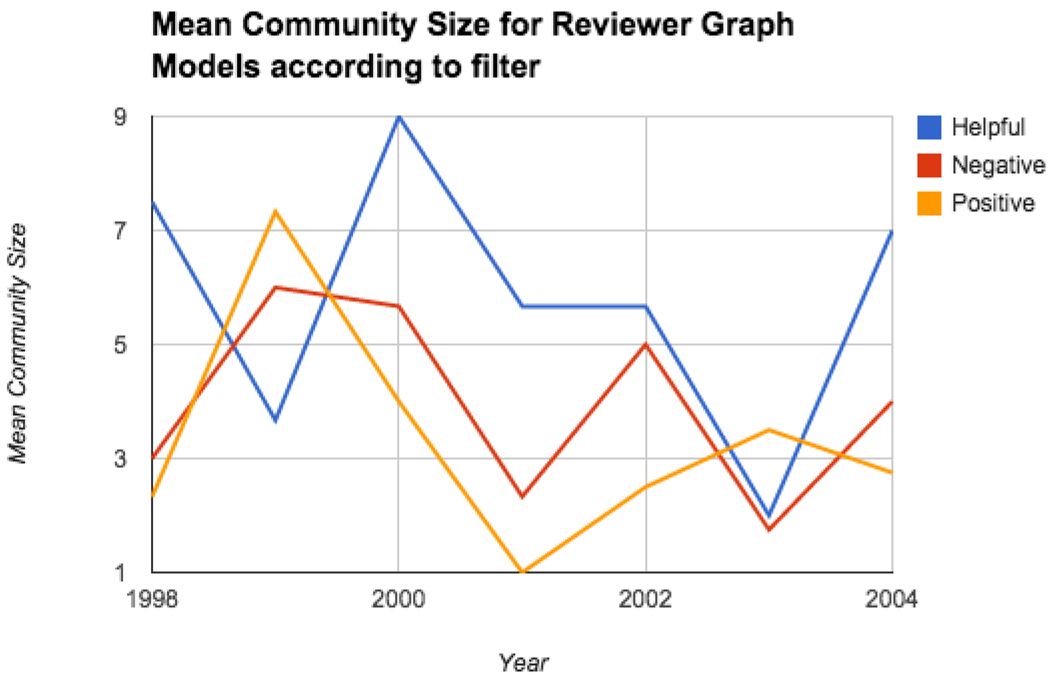
Quantitative Metrics of Communities

We explored two different quantitative metrics to try to understand the communities found by our algorithm. First, we examined the average size of the communities for each year. Second, we looked at the median salesrank of representative item set in a community.

Community Size

The community size is an approximation for how related the users in it are. Very large communities represent users who may be related by some general item category or very popular item, but users in that community are unlikely to be similar otherwise. On the other hand, communities of just a couple users likely arise from users having reviewed only a couple of items, making them disconnected from the rest of the graph.

Figure 2: the mean community size among all communities found by Infomap within the various reviewer graph models generated by three different filters



From the figure, we can see that we were able to find many communities with relatively small sizes. We found that the majority of communities the algorithm found were size 2 or 3, suggesting a heavy-tail distribution. This is likely due to the majority of reviewers in our graph reviewing only a few items, meaning the algorithm could not group them together with any other users.

Despite this, the algorithm was able to find a good number of larger communities. We observed that about 20% of the communities had at least 10 reviewers. Further, only about 10 communities had more than 250 reviewers.

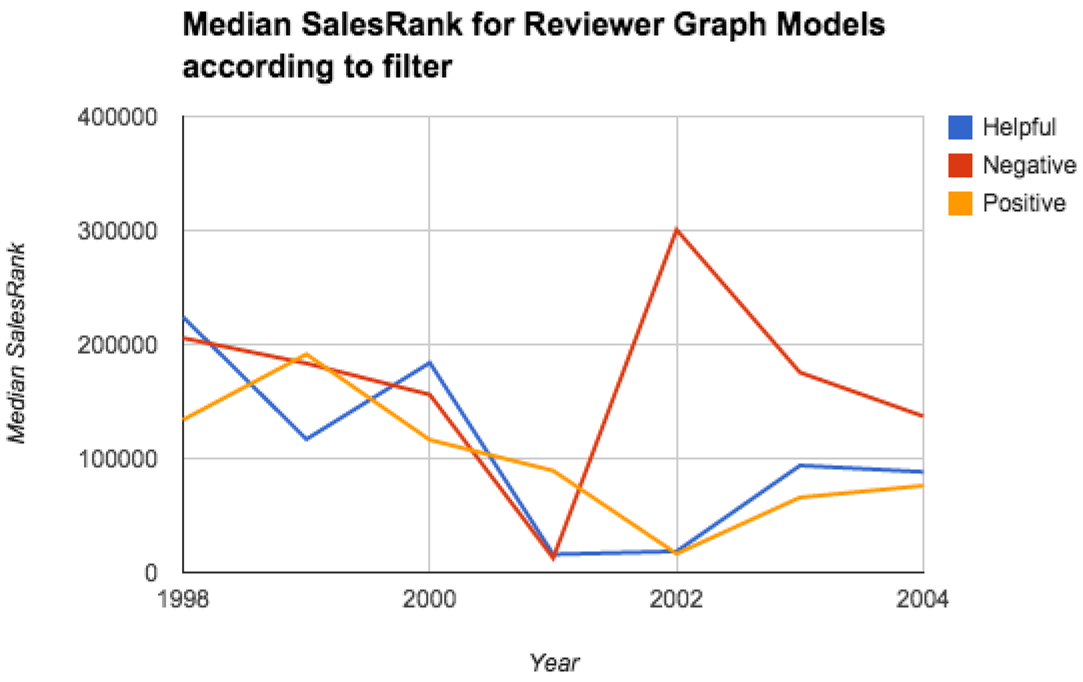
This suggests that many reviewers are grouped into small but significant communities. We expect these communities to be represented by a small set of similar items. As a result, we expect the reviewers in each community to be quite similar as well.

SalesRank of Representative Item Sets

The SalesRank of an item is a measure of how many people have bought the item. A low SalesRank indicates a very popular item, while a high SalesRank indicates a lesser-known one.

We analyzed the median SalesRank of the representative items for the largest community of each year. We chose the largest community because we expect that it corresponds to a broad collection of representative items. We wanted to see whether the item set we found was more or less popular depending on the model we used.

Figure 3: the median SalesRank for products that characterize the largest community found by Infomap within the various reviewer graph models generated by three different filters



From the figure, we can see that representative items under the negative filter are much less popular than under the positive filter. This is reasonable because the items in the negative filter community have received poor reviews from many users.

The rather low SalesRank of items in the positive filter community suggests that this community can be represented by a collection of popular items, whether or not these items are similar to one another, We would thus expect that the reviewers in this community are related by their interest in popular culture and trending products.

Another interesting aspect of the figure is that the SalesRank decreases over times. In the late 1990's, we observed that there was relatively little data, and even the largest community was represented by very few items. As such, that community was not characterized by popular items, but likely by more topically similar items.

In later years, with many more items and reviews, the largest community was generally represented by over a hundred items. The low SalesRank of these items suggests that reviewers who reviewed many popular items were all assigned to this community.

Description of Top Communities

In addition to the metrics above, we wanted to precisely characterize the representative itemsets of the top communities. Looking at the titles of these items, we saw a number of similar titles, but we could not find a clear pattern of similarity. Instead, we used item categories included in the dataset to try to find a more general pattern.

Below is a summary of the key words that appeared most frequently for each of the top five communities. Because items were generally assigned to multiple categories, there was a considerable amount of redundancy in the categories that we saw, which we have filtered out of the table.

Table 2: Description of the top five communities with two different filters for 1999 and 2003.

	Positive Ratings	**Negative Ratings**
1999	1. Books, Children's Books, Science Fiction & Fantasy, Harry Potter (series) 2. VHS/DVD, Drama, Action & Adventure, Science Fiction 3. Books, Mystery & Thrillers, Suspense, Horror, Religion & Spirituality 4. Books, Literature, Teens, Jane Austen 5. Books, People & Places, Friendship, Fiction	1. Books, Mystery & Thrillers, Suspense, Technothriller 2. DVD, Drama, Comedy, Horror 3. Books, Thrillers, Women Sleuths, Romance 4. Books, Literature & Fiction, Contemporary 5. Books, Science Fiction & Fantasy, Magic & Wizards, Sword of Truth (series), Wheel of Time (series)
2003	1. Books, Science Fiction, Literature, Classics, Religion & Spirituality 2. Books, Mystery & Thrillers, Contemporary, Suspense, Biographies & Memoirs 3. DVD, Drama, Action & Adventure 4. Music, Hard Rock & Metal, Alternative Rock, Classic Rock 5. Music, Alternative Metal, American Alternative	1. Books, Science Fiction & Fantasy, Epic, Wheel of Time (series) 2. Music, Hard Rock & Metal, Rap & Hip-Hop, Alternative Rock 3. Books, Nonfiction, Politics, Health, Popular Culture 4. Books, Mystery & Thrillers, Biographies, Military & Spies 5. Books, Literature & Fiction, Children's Books, The Chronicles of Narnia (series), Harry Potter (series)

We can see from the table that the communities we found were indeed represented by a particular category or genre of item. We found relatively little overlap in categories across communities, suggesting that this characterization of the communities is quite accurate.

We also see that the categories from positive filter communities rather accurately capture popular book series and genres of each year. *Harry Potter*, for example, showed up many times in the largest community of 1999--this seems to be the reason for the other keywords in that community, like "Children's Books," under which it was categorized.

Similarly, the categories of negative communities accurately represent controversial genres or books and movies that generally received mixed reviews. The second community of 2003, for example, appears to have identified the group of reviewers who dislike rap, hard rock, and metal, genres which were quite popular among younger generations at the time.

This result above all shows that our models and chosen algorithms were able to effectively group users according to the products they liked or disliked.

Conclusions

Accurately characterizing Amazon reviewers is difficult because we do not have any information about the users themselves other than their product reviews. As such, we explored a variety of models, algorithms, and filters to define the reviewer network and extract information and meaningfully categorize reviewer communities.

With these models, we looked beyond simple graph properties such as strongly connected components to find communities. By using InfoMap, we were able to find groups of reviewers who were clearly related by a set of similar items or topics.

References

[1] A. Clauset, M.E.J. Newman, C. Moore. *Finding community structure in very large networks.* Phys. Rev. E 70, 066111

[2] , 2004.

[3] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, L. Lee. *How opinions are received by online communities: A case study on Amazon.com helpfulness votes*. In Proc. ACM WWW, 2009.

[4] D. Shinzaki, K. Stuckma*in Amazon Reviews: Final*

[5] Edler, Daniel, and Martin Rosvall. "Mapequation.org - Code." *Mapeq*2014. Web. 3 Dec. 2014.

[6] Leskovec, Jure. "Network Community Detection: Modularity Optimization and Spectral Clustering." 13 Nov. 2014. Lecture.

[7] Leskovec, Jure. "Link Analysis: HITS and PageRank" 4 Nov. 2014. Lecture.