Personalized Product Recommendations

Prasun Kanti Dey, Emily Hok, Rajat Bhattacharjee

December 7, 2017

Abstract

This paper presents the Personalized Product Recommendation (PPR) system which intends to find the top recommendations for a current product. PPR utilizes PageRank algorithm on Amazon dataset for ranking the products and lists the top ranked products based on similar category or group or combining products from all categories.

1 Introduction

Online shopping is becoming increasingly popular. Furthermore, it comes as no surprise to learn that the popularity of online consumer reviews is also increasing tremendously. A recent study revealed that online consumer reviews are the second most trusted form of advertising, the first being recommendations from friends and family [Gri12]. Additionally, about 88% of Americans are internet users as of June 2017 [int17]. With the rise of internet users, the user's ability to shop online has also improved significantly. An example of such improvement helped pave the way for e-commerce giants like Amazon, whose remarkable success is partially attributed to its recommendation algorithm to personalize the online store for each customer [LSY03].

To effectively develop personalized product recommendations, determining the predictors of helpfulness of online reviews is important. Understanding the relationship between product reviews and sales would be highly beneficial as a recent research report revealed that 92% of online consumers read product reviews before making a purchase decision [ema10]. Previous studies conceptualized review helpfulness as a voting ratio by consumers [FGW08]. However, an inherent problem to consider if solely relying on a voting mechanism is the possible bias that could occur. For an example, reviews that accumulate votes receive more attention than the less voted reviews, which perpetuates the standing of the "most helpful" review. This phenomenon is known as the winner circle bias [LHTW13]. Furthermore, for the purpose of this study, we conceptualized review helpfulness as the subjective views of the consumers' perceived source credibility of the review. In other words, if half or more the consumers that voted on a review deemed it helpful, our algorithm will label the review as "helpful".

The next objective is to process the final network and compare our results to the similar items fields in our dataset. We observed the structure of the network of products is similar to the link structure of the World Wide Web. Thus, we decided to utilize the PageRank algorithm on our network of products to determine the most popular items. PageRank analyzes the link structure, that is, the forward links (outedges) and backlinks (inedges) of the web and produces a global "importance" ranking for each web page. The rank of the page is divided among its forward links so that they evenly contribute to the ranks of the pages they point to. The computation continues until it converges [PBMW99]. While the product's category is a strong determinant in the recommendation procedure, we can sort the list of items in the same category by its PageRank in descending order and recommend the products at the top of the list.

Moreover, the primary aim of this paper is to develop a personalized recommender system for online reviews. The algorithm creates the network based on the voting mechanism to provide a binary classification of whether the review is helpful or not helpful. It then sorts the products using the PageRank algorithm and then chooses the top products based on its global ranking. Results from the proposed recommender system can then be compared to the results of Amazon's similar products found in the dataset. The remaining parts of the paper are organized as follows. We start with how we formulated and how our proposed recommender system works. We then proceed to the results of the study. Lastly, we discuss the study's potential contributions as well as its limitations.

Table 1: Amazon product co-purchasing network metadata

Dataset statistics	
Products	548,552
Product-Project Edges	1,788,725
Reviews	7,781,990
Product category memberships	2,509,699
Products by product group	
Books	393561
DVDs	19828
Music CDs	103144
Videos	26132

Table 2: Sample product format from dataset

Id: 10
ASIN: 0375709363
title: The Edward Said Reader
group: Book
salesrank: 220379
similar: 5 039474067X, 0679730672, 0679750541, 1400030668, 0896086704
categories: 3 |Books[283155]|Subjects[1000]|Literature & Fiction[17]|History & Criticism[10204]|Criticism & Theory[10207]|General[10213] |Books[283155]|Subjects[1000]|Nonfiction[53]|Politics[11079]|History & Theory[11086] |Books[283155]|Subjects[1000]|Nonfiction[53]|Social Sciences[11232]|Anthropology[11233]|Cultural[11235]
reviews: total: 6,downloaded: 6,avg rating: 4 2000-10-8 cutomer: A2RI73IFW2GWU1, rating: 4, votes:12, helpful: 7 2001-5-4 cutomer: A1GE54WF2WUZ2X, rating: 5, votes: 11, helpful: 8 2001-8-27 cutomer: A36S399V1VC4DR, rating: 4, votes: 5, helpful: 3 2002-1-26 cutomer: A280GY5UVUS2QH, rating: 3, votes: 12, helpful: 7 2004-4-7 cutomer: A2YHZJIU4L4IOI, rating: 4, votes: 10, helpful: 2 2004-4-27, cutomer: A1MB83EO48TRSC, rating: 4, votes: 5, helpful: 3

2 Methodology

2.1 Dataset

The dataset used in this paper was acquired from Stanford Network Analysis Project [sna] (presented in Table 1). Components found for each in the dataset are the ID (the product's position in the dataset), an Amazon Standard Identification Number (ASIN), a title, a group label, sales rank, list of similar products, list of categories that the product falls under, total reviews by the users who purchased this product, and average rating. The reviews can be further analyzed as it provides details of when the product was purchased, the unique customer ID of the purchaser, the rating given by the customer, the total number of people who voted on the review (which included both helpful and not helpful votes), and the number of people who found the review helpful. A sample product detail is shown in Table 2.

2.2 Predictive Model

The central idea of this study is to develop product recommendations based on the product's category and product reviews, which will produce a value in a created network that represents the likelihood of a product being recommended. The system can then find the list of products with higher probability of purchase by the user and sort the values in descending order with the products that has the largest values on top and the products with the lowest values on the bottom.

2.2.1 Dataset Reduction

The first step in our methodology procedure is to reduce the dataset. To accomplish this, products with less than three reviews were discarded. The reasoning behind this is to ensure the connectedness of the network of products we want to create. Products with one or two reviews offer very few links and are also expensive to process if there are too many of them. Another criterion to reduce the dataset is to discard products that have no helpful votes. The logic behind this implementation is to increase the likelihood of recommending high quality products. By checking the quality of the product reviews, we are assuring the credibility of the product rating. Finally, to further increase the likelihood of a connected network of products, we included items from a sorted list of top buyers until the product list reaches 4,000.

2.2.2 Identifying Helpful Review

The next step is to formulate an equation to determine the likelihood of a product being recommended.

Helpful Ratio We define helpfulness of a review as:

$$Helpful\ ratio = \frac{Number\ of\ helpful\ votes}{Total\ number\ of\ votes}$$

Total number of votes includes both helpful and not helpful.

If the helpful ratio is equal to or greater than 0.5, we classify the review as helpful. On the other hand, if the helpful ratio is less than 0.5, we classify the review as not helpful and discard the review so that particular review will not affect the overall product rating.

After discarding the reviews based on their helpful ratios, we multiply the helpful ratio with the rating for each review. We then take the sum of the product of the helpful ratio and rating and divide that value by the sum of the helpful ratio.

$$product\ rating = \frac{\sum_{each\ review} helpful\ ratio * rating}{\sum helpful\ ratio}$$

However, we normalize the value by dividing it by 5 as each product has a rating of $[0, 5]$ in Amazon dataset.

Example Our overall procedure can be listed as following (based on example in Table 2):

1. Discard review "2004-4-7 customer: A2YHZJIU4L4IOI rating: 4 votes: 10 helpful: 2" (helpful ratio = 0.2)

2. Multiply helpful ratio with product rating for remaining reviews

$$0.583 * 4; 0.73 * 5; 0.6 * 4; 0.583 * 3; 0.6 * 4$$

3. Divide sum of products by sum of helpful ratios

$$\frac{12.52}{3.096} = 4.04$$

4. Normalize value to get the final rating

$$\frac{4.04}{5} = 0.808$$

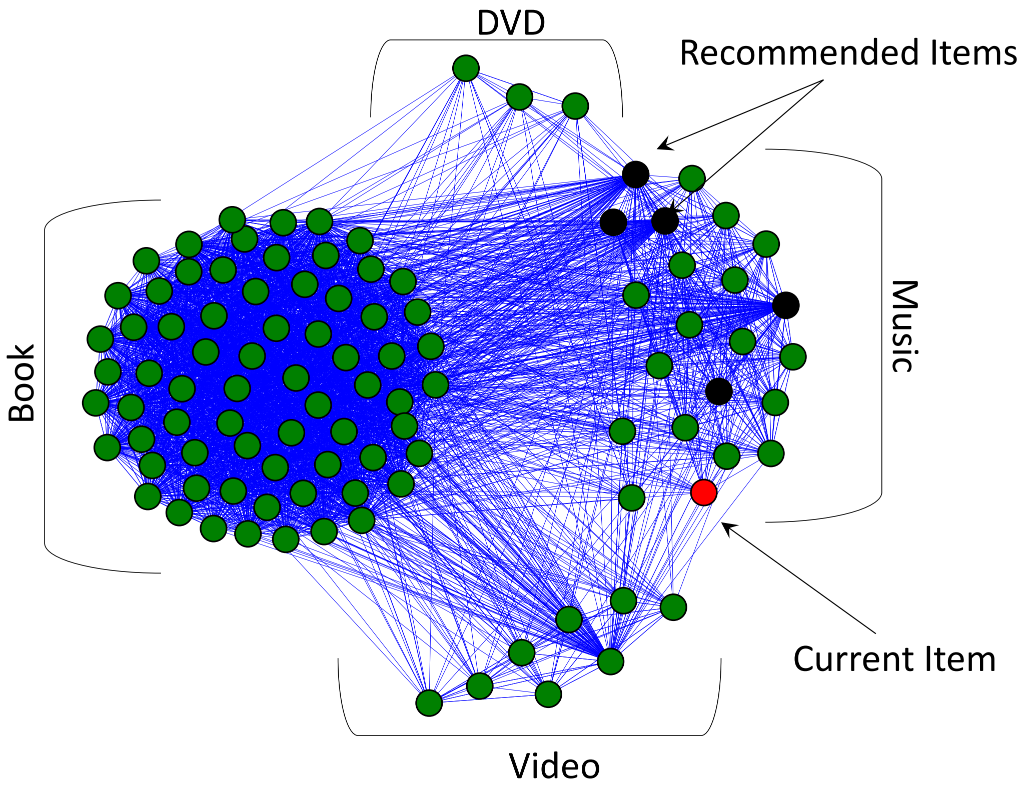

Figure 1: Sample Graph with 102 nodes

2.2.3 Creating Edge

Finally, we create a directed graph as relations between two items that are being considered. Based on our observation and to reduce the number of edges (to make more precise recommendation), we used some threshold for creating an edge between two nodes. In general, if two nodes A and B belong to the same category (i.e, Jazz in Music or Mystery in Book), an edge will be created between them. However, we do not want to create many edges as some of the items have very poor helpful ratio. If helpful ratio > 0.7 for similar category items, we put an edge between them. We also created edges between items from different categories but in the same group (i.e, Music, Book or Video) with helpful ratio threshold > 0.8. Following this methodology, the graph we constructed is a clustered graph, as edges only exist between same category or group items. This means, we will always recommend products from the same group or category. To avoid this, we also create an edge from A to an item B that has very high helpful ratio (0.999) regardless of their groups.

It is important to note that the edge calculations described here are solely based on the product at the head of the arrow. In other words, the calculation for an edge between product A and product B signifies the likelihood of product B appearing on web page of product B.

2.2.4 Assign Weights on Edge

In general, edges are given weights based on the head node's helpful ratio. The original PageRank assumes the same weight for all of its incoming edges for each node, thus, modifications were made. We put weights on each of the edges based on the relationship between the nodes. For example, if both the nodes are from same category, we keep the edge weight same. If edges connecting nodes are from a similar group, then we divide the weight by two, while other edges have a weight of one-third of their original values. This is done to ensure that products from similar categories are prioritized first before searching other items while recommending.

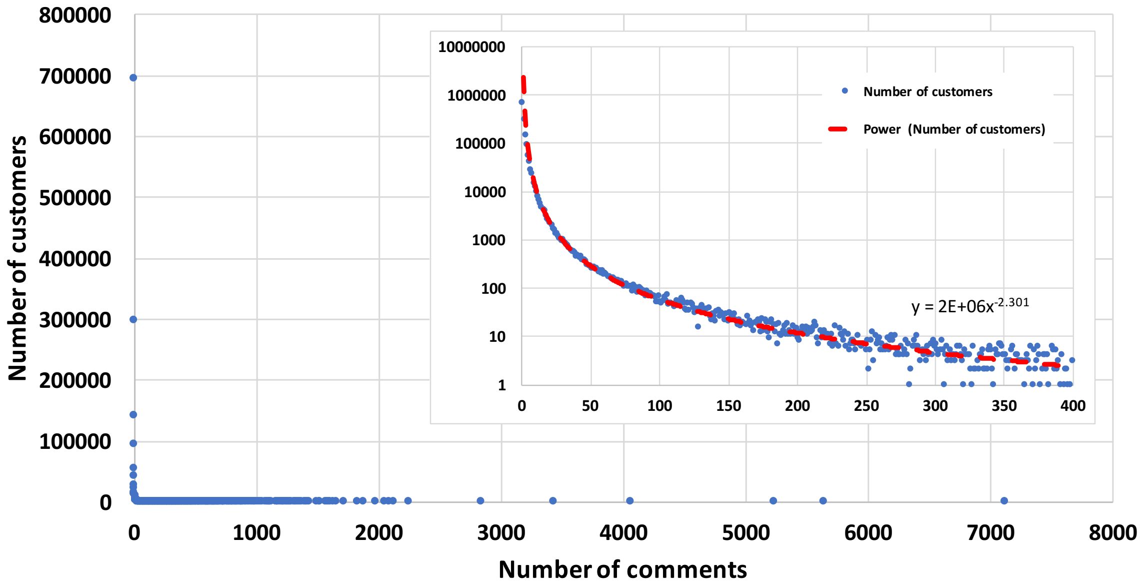

Figure 2: Customer comments

3 Experiment

3.1 Setup

We used Python version 2.7 [pyt] for our experiment since python can handle large data easily and provides better convenience. Python library *re* was heavily used as we have to parse the dataset and find each item and their related informations. DataFrame from *Pandas* library [pan] was used for storing temporary data while reducing the dataset, and for graph construction and all of the operations on it was managed using *Networkx* library [net].

Figure 1 shows a sample graph constructed from 102 nodes and 4938 edges. Red node represents the current product for which we want to find the recommended (Black nodes) products. As one can see, the graph is clustered and separated into four groups (book, music, video and dvd)

3.2 Observations

We made some interesting observations from the dataset. While reducing the dataset, we identified the top users who reviewed (commented) the most. Figure 2 shows the number of reviews made by users. We found that number of reviews followed a power law distribution with $\alpha = 2.301$.

Almost 700,000 user reviewed at least one product and less then 10,000 users reviewed 11 products. We discarded two users with reviews greater than 200,000 and 900,000, respectively. We are not certain about the content or the context of the discarded reviews. However, because both users commented multiple times on the same product, we assumed the users were replying to their own review or answering another user's question.

3.3 Results

We ran two experiments with different dataset sizes. The first experiment contained 102 products and the second contained 4000 products. Due to our helpfulness and edge criteria, the second experiment contained 3980 products, instead of 4000, with 7278561 connectivity among the products.

For recommending items, we prioritized as we discussed before. The first method used to fill the recommendation list was to select products of the same exact group. If we could not find enough products, we searched for products belonging to a similar group. Finally, if that was not enough, we added products from the top of the PageRank list.

Figure 3 shows a sample output of our recommendation system. As one can see, it gives three types of recommendations, which were directly from similar category, from similar group and from all categories. The last one is the final that uses the prioritized order.

```
Top 5 Recommended Items for current item ASIN: 0767905288:
From same Category "Nonfiction[53]                              " are: ['0760312087',
'0253210674', '0882951254', '0465098290', '0760301352']
From same Group    "Book                                        " are: ['0931948320',
'0452280400', '019506075X', '0521565812', '1579890164']
From all categories are                                           : ['0931948320',
'0452280400', '019506075X', '0521565812', '1579890164']
Combined top 5 recommendations (category > group > other)         : ['0760312087',
'0253210674', '0882951254', '0465098290', '0760301352']
Amazon suggested similar                                          : ['0767908481',
'0767913795', '0060514558', '0060582510', '0060544244']
```

Figure 3: Sample Recommendation

```
Top 5 Recommended Items for current item ASIN: B0000262WI:
From same Category "Jazz[34]                                    " are: ['B000003DGW',
'B0000027K0', 'B000005IZH', 'B00000363W', 'B000009RML']
From same Group    "Music                                       " are: ['B000000DYZ',
'B000005KRA', 'B000003QH2', 'B000004BTK', 'B000002L4I']
From all categories are                                           : ['0931948320',
'0452280400', '019506075X', '0521565812', '1579890164']
Combined top 5 recommendations (category > group > other)         : ['B000003DGW',
'B0000027K0', 'B000005IZH', 'B00000363W', 'B000009RML']
Amazon suggested similar                                          : ['B000025ZP8',
'B00002EPJH', 'B0000247YQ', 'B0000261GX', 'B000024SN9']
```

Figure 4: Sample Recommendation (2)

We want to highlight two things here, *i)* it seems 'same category' and 'combined top 5' produces the same recommendation. This is expected as 5 products have been found in similar categories, so it will return them as best recommendation. *ii)* 'same group' and 'all categories' also produces the same list. This is because our dataset consists mostly of books and have very high helpful ratios. Moreover, when we created the network and calculated PageRank, these books achieved the highest rank and, thus, consistently got recommended. This issue is shown in Figure 4. Here, the product belongs to "Jazz" category in Music group, but still has the same recommended items list in 'all categories' type.

```
Top 5 Recommended Items for current item ASIN: 0790747324:
From same Category "Science Fiction & Fantasy[163431]  " are: []
From same Group    "DVD                                " are: ['B00000IC8Z', 'B000056PNB']
From all categories are                                  : ['0590568833', '0613100093',
'1563824035', '0834210126', '014023778X']
Combined top 5 recommendations (category > group > other)   : ['B00000IC8Z', 'B000056PNB',
'0590568833', '0613100093', '1563824035']
Amazon suggested similar                                 : ['B00007JMD8', '6305350221',
'B00004RF9B', 'B00005JKFR', 'B00005NG6A']
```

Figure 5: Recommendation with fewer similar category products

Since the second experiment has 4000 products, most of them will have at least five products from same category. However, to show that our system works, we present an example from the first experiment for product ASIN: 0790747324, which is the only product from "Science Fiction & Fantasy" category and has only two products in its "DVD" group. As a result, the final recommendation takes the first three products from other groups (Figure 5).

3.4 Accuracy

Our system assumes that all product reviews given by the customers are legitimate and customers actually purchased the products. To verify our system's accuracy, we compared our system's predicted list with the similar five items that was already provided by Amazon in the dataset for each product. Unfortunately, we do not know how Amazon actually recommends items nor do we have all of the products from Amazon's inventory. Furthermore, it is expected that not all of the

```
Top 5 Recommended Items for current item ASIN: 0790747324:
From same Category "Science Fiction & Fantasy[163431]  " are: ['B00005LC3X', '6305350221',
'B00005JKHP', 'B00003CXFG', 'B00003CY5D']
From same Group    "DVD                                " are: ['B00004TX05', 'B00005YUOC',
'B00006JDUW', '0792299922', '0780022343']
From all categories are                                  : ['0931948320', '0452280400',
'019506075X', '0521565812', '1579890164']
Combined top 5 recommendations (category > group > other)   : ['B00005LC3X', '6305350221',
'B00005JKHP', 'B00003CXFG', 'B00003CY5D']
Amazon suggested similar                                 : ['B00007JMD8', '6305350221',
'B00004RF9B', 'B00005JKFR', 'B00005NG6A']
Items matched with our recommendation with Amazon: set(['6305350221'])
```

Figure 6: Recommendation matchs with Amazon recommendation

items listed in the "similar five items" will be in our dataset, hence, the system's very low accuracy when compared to the similar five items provided by Amazon. Needless to say, considering the dataset, the products being recommended for a certain product have actually good helpful ratio with maximum number of reviews in that category.

We ran our second experiment more than 100 times. In the experiment, five random products were chosen as the current product, and then for each product, five products were recommended for each of them. Only one item was found to be a match. As we can see from Figure 6, our systems recommends product ASIN "6305350221" for "0790747324", which is also recommended by Amazon. This proves that if we run our system with a larger dataset, we will find more matches.

3.5 Time complexity

Dealing with big data means we have to carefully write our code so that it can optimize space and run time. Table 3 shows details of time complexity of two experiments that we ran.

As we can see from the table, sorting the items based on Pagerank is the most time consuming step (0.15 sec in Ex1 and 357 sec in Ex2). Edge creation also takes a significant amount of time due to our algorithm having a runtime of $O(n^2)$.

Table 3: Time comparison for two experiment (in seconds)

	Experiment 1	Experiment 2
Product (Node) count	102	3980
Connectivity (Edge) count	4938	7278561
Graph Creation time	0.053941	1.553151
Edge Creation time	0.05423	73.436685
PageRank calculation time	0.156781	357.741793
Time for Top 5 recommendation		
Product 1	0.000173	0.044313
Product 2	0.000218	0.003178
Product 3	0.000252	0.045157
Product 4	0.000236	0.007857
Product 5	0.000122	0.007994
Total time elapsed	0.266745	432.848794

4 Discussion and Conclusion

In this project, we first reduced the *Amazon product co-purchasing network metadata* dataset from 548,552 records to 4000 records to ease our implementation. Otherwise, the program became unresponsive due to its computational overload. Based on our results, we found that reducing the dataset significantly affected the accuracy since most products were eliminated. Additionally, it is possible we discarded some helpful reviews by only keeping products purchased by top buyers and discarding the rest. Nevertheless, out of those reviews, we only kept the products that had more than 3 reviews and received a good amount of helpful votes.

We then developed a network of product nodes extracted from our reduced dataset. We iterated over all possible combinations of nodes and calculated a weight value derived from our weight equation. This weight equation took into consideration the number of helpful reviews and its rating. We then calculated our weight for that Amazon product. This careful calculation of weight ensured the accuracy of good reviews. If this normalized weight was more than 0.7, we created a directed edge between the two product nodes.

After constructing our graph with all nodes and edges, we implemented a PageRank algorithm to find the most recommended items. To test its accuracy, we compared it to similar items from that specific product record in the original dataset.

The categories used to determine similar products were also a limitation for this study. Because categories were generic by nature due to the time constraint, sub-categories were not at all considered. Hence, to optimize this program, implementing a function that can process more subcategories should yield more accurate results. This is something we can do in future. Also, we derived our set of users from the ones mentioned in the reviews. Users that were not present were not included in our user list. We can also implement *Collaborative filtering* [SKKR01] to compare recommended list with our approach.

In addition to this, one of the crucial components of the algorithm relied on what we defined as 'helpful' reviews. Since the predictive results were of binary values only, helpful or not helpful, the algorithm was then designed to perform with only 50% accuracy. Moreover, if the recommender system produced results with accuracy greater than 50%, we considered the algorithm's predictive ability to be more accurate.

References

[ema10] The Role of Customer Product Reviews. Emarketer, 2010. Available online: https://www.emarketer.com/Article/Role-of-Customer-Product-Reviews/1008019.

[FGW08] Chris Forman, Anindya Ghose, and Batia Wiesenfeld. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information Systems Research*, 19(3):291–313, 2008.

[Gri12] Marisa Grimes. Nielsen: Global Consumers' Trust In 'Earned' Advertising Grows In Importance. Nielsen, 2012. Available online: http://www.nielsen.com/us/en/press-room/2012/nielsen-global-consumers-trust-in-earned-advertising-grows.html.

[int17] Internet Usage Statistics: The Internet Big Picture. Internet World Stats, 2017. Available online: http://www.internetworldstats.com/stats.htm.

[LHTW13] Mengxiang Li, Liqiang Huang, Chuan-Hoo Tan, and Kwok-Kee Wei. Helpfulness of online product reviews as seen by consumers: Source and content features. *International Journal of Electronic Commerce*, 17(4):101–136, 2013.

[LSY03] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[net] Networkx: Software for complex networks. Available online: https://networkx.github.io.

[pan] Python Data Analysis Library. Available online: https://pandas.pydata.org.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[pyt] Python 2.7. Available online: https://www.python.org.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[sna] Amazon product co-purchasing network metadata, key = , month = , year = 2006, note = "available online: https://snap.stanford.edu/data/amazon-meta.html",.