CPT_S-591
Elements of Network Science
Project Report

Team:12
Zain Mudassar
Inshal Navqi
Pradnya Nimbekar

Optimization Techniques for Recommendation Systems

Introduction:

In the past couple of years, the information technology space has been concentrating highly on data and how to make it sense of all the data that is being gathered. Living in the age of information overload we face so much information that it becomes problematic to make a concrete decision. Therefore, we tend to find definitive ways that may help in alleviating this problem and help us make sensible decisions in most of the cases. For example, people ask their friends, read newspapers, advertising, general surveys, and so forth to get knowledge about items to make their decision easy. Nevertheless, people still suffer from information flooding which progressively is becoming a big challenge in their daily life. This encourages more and more researchers to develop new techniques that help users' in dealing with this issue more easily and effectively. Because of this, researchers introduced a phenomenon called the “Recommendation System”.

The fundamental way this works is that these systems rely on the history of activities from the users which is utilized to build a user profile about their preferences. These systems enable users of the internet to find their preferred information from the massive pool of the data which is available on the internet in a quick and efficient fashion. These researchers develop techniques which provide us with tools, their aim is to predict rating or preference that a certain user would assign to an item not yet considered such as a new book, movie, or a new product. Making smart use of these prediction values, the system can then help users find items they prefer. These recommendation systems use a variety of features to make these predictions like, using the content or features of the item (content-based approach), feedback information given by users on items (collaborative filtering) or a combination of both approaches (hybrid approach).

Collaborative Filtering algorithms predict a rating for an item based on the user's previous ratings for other items as well as rating other users. Most Collaborative Filtering systems provide simple single point predictions of the ratings. To evaluate the accuracy of those predictions, a simple metric (such as mean absolute error or mean squared error) is applied that compares predicted ratings to the actual ones. Some accuracy metrics were also proposed for situations when the recommender system serves a different purpose such as selecting N best items or based on utility of the recommendation such as revenue of the company resulting from the recommendation. Some other metrics loosely related or not related at all to prediction accuracy were also proposed to evaluate other aspects of CF systems such as coverage, novelty, or serendipity. Please note that while other aspects of the recommender system are considered, most of the evaluated systems still offer only the point predictions of the user ratings.

To suggest relevant items to users, there exist two major categories of methods, content based and collaborative filtering methods. Content-based recommendation systems recommend items to a user by using the similarity of items. It identifies the similarity between the products based on their descriptions. Collaborative filtering uses similarities between users and items simultaneously to provide recommendations. It further categorized into two classes: rating-oriented and ranking-oriented algorithms. Rating-oriented algorithms have the goal of reliably predicting a user's ratings and then recommending the products with the highest predicted rating for him. Ranking-oriented approaches, such as collaborative ranking, aim to predict item rankings directly from the perspective of a target user without specifically predicting the rating. Users are selected for their similarity to active users, this similarity is determined by matching users who have posted similar reviews, based on previous similarity it is assumed that future dislikes and likes will be similar. From the average group, recommendations are made for the active user this is known as neighbor-based filtering. In item-to-item based filtering a matrix

is used to determine the likeness of pairs of items. After this we compare the preferences of the current user to the items in the matrix for similarities through which we can make recommendations. A classification based collaborative filtering system recommends things based on how similar users liked that classification. It is assured that users who enjoy or dislike similar experiences within a classification will also enjoy other things in that same classification.

Some collaborative filtering systems are memory-based, like neighboring- and item-to-item models, which compare similarities of users or items. Others are model based, using machine learning to compare dissimilar items. Model-based systems may use algorithms such as the Markov decision process to predict ratings for items that have not yet been reviewed. Hybrid systems include features of both memory-based and model-based filtering. Recommendation systems are used to provide suggestions for all kinds of websites and services. Still, they can encounter several difficulties. The sparsity of ratings is one of the main hurdles to collaborative filtering's usefulness in systems with many items. New items also tend to be difficult to provide recommendations for. Under new recommendation systems, it is hard to provide good recommendations before enough users have entered reviews. At the same time, however, too many user ratings can be challenging to some systems because they make for huge data sets.

Motivation:

Our goal behind this project was to find ways to improve collaborative filtering results when there are not enough co-rated items to depend on. If there are no co-rated items, similarity computation cannot be performed, this is known as co-rated items problem. We certainly believe collaborating filtering techniques have been made sufficiently fast and scalable. Instead, our concerns pertain to the accuracy of the predicted ratings given the inherently sparse data. Equivalently, we believe that these algorithms require more user ratings than necessary to make accurate predictions, thus phrasing our concerns in the dual context of a learning curve instead. Note that this ability to make accurate ratings predictions in the presence of limited data is by no means a trivial issue: Collaborative filtering algorithms are not deemed universally acceptable precisely because users are not willing to invest much time or effort in rating the items. Unquestionably this issue is the Achilles heel of collaborative filtering.

Collaborative filtering is mostly useful on the very versatile platform of e-commerce and increasing enough we see it being used in web streaming services but is mostly for websites selling homogeneous items. Examples of such commodities include books, video games and even software, many other applications are possible. For example, collaborative filtering could be used to rate web pages themselves. In this specific case, users spend time on a web page serves as surrogate for the rating. Another concept we see growing is personalization of web pages according to the user, on a side note we believe that this is an extremely important e-commerce opportunity. Inherently the goal for collaborative filtering algorithms is to be able to make accurate rating predictions, though we believe ranking to have a slight edge depending on the use case, to do so in a manner which is fast and scalable.

Problem Definition:

In many applications, all we have is a set of implicit feedbacks while no rating data is available and hence, rating based methods cannot be used in such a situation. Implicit feedback can be automatically gathered by tracking the user's interactions with the system. It has been shown that ranking-oriented collaborative filtering approach is sometimes more intuitive and applicable. Common approach is Neighbor-based collaborative filtering, one of the main classes of collaborative filtering, estimates the

ranking/rating of target user based on the behavior of similar users. However, this is not much effective. This is because in recommender systems, users have given feedback to a small proportion of items, and consequently, they rarely have enough common items or pairwise comparisons for estimation of their true similarities/ dissimilarities.

One optimized approach to overcome this issue, is graph-based recommendation that takes advantages of heterogeneous information networks, that are information networks containing different types of nodes and edges, to refine the similarity measures. Graph-based recommendation algorithms are composed of two steps: Constructing a graph representing the data and making recommendations by analyzing the graph. Also, we have used concept of ego network while doing the analysis.

Model/Algorithm:

We propose a graph-based model based on ego-network and Island method. We have two major parts of our model, pre-analysis in which a graph is constructed representing the data and other part is to provide recommendations by analyzing the graph. For analysis, we have majorly used ego network and island method. The island method is particularly well-suited to valued networks. To understand the island method, we visualize an island with a very complex terrain where the height of each of the peaks are defined by the value of the node or edge. Now assume that the water level around this island begins to rise slowly, eventually water is going to leave portions of the landscape of this island underwater. Those peaks we discussed earlier form valleys, when these valleys are flooded, as the water level slowly rises the island. Now the island is split into a bunch of smaller islands which makes it possible to clearly identify the high peaks on the island and as the water level keeps on rising making these peaks smaller. In a careless application the water level may rise high enough that the entire island disappears. So. this method needs to be applied carefully so that it can effectively show meaningful results.

To understand the ego network, consider local networks with one central node, the ego, are known as egocentric networks. The ego is at the center of the network, and all other nodes directly linked to the ego are referred to as alters. During data collection and analysis, an Ego is the network's focal point and is surrounded by alters. Since the ego is the only one that can nominate an alter, egocentric networks are often referred to as "perceived" or "cognitive" networks. Person links for social support, access to services, knowledge dissemination, and changes as prominent actors are studied using ego network research. For example, in the Amazon data set, one product which could be a center node is connected to 5 other products which will be the alters of ego node/ center node.

Based on the product id our model will obtain the metadata about the product and then get degree-1 ego network by taking the product that has been co-purchased with this product earlier. Then identifying the most similar products using Island method with threshold for edges >=0.5 as it will give the most similar products. This is how we get our final output.

Results and Findings:

The fundamental of recommendation systems is finding attributes with identical properties and checking the similarity. As discussed above, our approach is to provide a graph-based recommendation based on a graph data model based on Ego Network and Island method. This solution is quick and effective because it is directly proportional to the action of the customer and evaluates their recommendation instantaneously.

Data Analysis:

For study, we have used the Amazon data set from snap.stanford.edu. The dataset contains product metadata and review information about 548,552 different products. It gives us enough knowledge to experiment with and come up with concrete findings and conclusions for the project.

For each product we have below information:

1. **Product ID:** Numeric values
2. **ASIN:** Amazon Standard Identification Number assigned by Amazon for product identification.
3. **Title:** Name of the Product.
4. **Group:** Product type, could be books, music CDs, DVDs and VHS video tapes.
5. **SalesRank:** Representation of the sales of that product compared to the others in its category.
6. **Similar:** ASINs of co-purchased products
7. **Categories**: Gives the specification of the product's category hierarchy, e.g., genre etc. (separated by |, category id in [ ]).
8. **Reviews:** Product's review information
9. **Total Number of Reviews**
10. **Average Rating:** Individual customer review with time, user id, rating, total number of votes on the review and the number of people who found the review helpful.

Graph structure for Amazon recommendation system is as follows:

- **Nodes** - ASIN which is Amazon standard identification number assigned for product identification number.
- **Edges**- Relation between two ASINs had an edge if they were co-purchased.
- **Edge weight:** This was determined based on the category similarity of the products. This measure of similarity is generated using the "Category" data, where Similarity is between 0 and 1 such that: 0 is the least similar and 1 the most similar.
- **Degree Centrality** -It is defined as count of the number of neighbors a node has.
- **Clustering Coefficient:** The degree to which nodes in a graph tend to cluster together.

Approach:

As the data has many records, we are only considering the category as Books and based on co purchased data, recommendation will be provided using social network analysis.
There are two steps in the implementation, first is pre-treatment to the data or pre-analysis of data and the second step is implementing graph-based algorithm using the Island method and ego network and applying it for recommendation.

[I] Pre-analysis (Filtering and Graph construction):

Implementation Steps:

1. Read data from amazon-meta.txt and populate amazon products nested dictionary.
2. Create books-specific dictionary exclusively for books.
3. Remove any copurchased items from copurchase list if we do not have metadata associated with it.

4. Create a product copurchase graph for analysis where the graph nodes for product ASINs and graph edge exists if two products were copurchased, with edge weight being a measure of category similarity between ASINs.
5. Get Degree Centrality and clustering coefficients of each ASIN and add it to amazonBooks metadata.
6. Write amazonBooks data to file and write copurchaseGraph data to file.

Output of pre-analysis:

0.7	0.6	0.64	0.44	0.58	0.7	0.88	0.48	0.88	0.62
0.5	0.33	0.64	0.39	0.1	0.69	0.88	0.39	0.76	0.9
0.8	0.78	0.44	0.68	0.8	0.46	1	0.42	0.63	0.9
0.8	1	0.78	0.94	1	0.47	0.79	0.48	0.88	0.9
0.7	0.45	1	0.44	0.7	0.43	0.1	0.16	0.85	0.86
0.7	0.9	1	0.21	0.7	0.64	0.87	0.32	0.69	0.9
0.7	1	0.5	0.33	0.58	0.57	0.48	0.31	0.55	
0.7	0.78	0.4	0.44	0.7	0.35	0.81	0.42	0.36	
0.7	0.64	0.23	0.14	0.28	0.53	0.71	0.26	0.79	
0.75	0.7	0.28	0.33	0.62	0.5	0.75	0.94	1	
1	0.78	0.44	0.39	0.47	0.73	0.19	0.73	0.9	
0.9	0.9	0.44	0.32	0.9	0.31	0.1	0.81	0.9	
0.78	0.9	0.47	0.39	0.9	0.17	0.34	0.59	0.9	
0.39	0.64	0.47	0.44	0.42	0.4	0.39	0.45	0.9	
0.5	0.6	0.26	0.83	0.58	0.57	0.41	0.25	0.9	
0.39	0.78	0.39	0.8	0.58	0.42	0.48	0.69	0.56	

Result Discussion of Pre-Analysis

In pre-analysis, first we read the data from amazon-meta.txt and then create Book specific dictionary as the data set is huge and cannot run so many records on local machine due to limited system configuration. Addition to that two more data columns added, one for degree centrality and other is for clustering coefficients. As discussed earlier, degree centrality is the count of number of neighbors a node has, and clustering coefficient is calculated as a degree to which nodes in a graph tend to cluster together. Lastly, a product copurchase graph is created for analysis where graph edge exists if two products where copurchased then copurchase graph data has been written to a file called amazon-books-copurchase.edgelist.

[II] Graph Based Approach:

Implementation Steps:

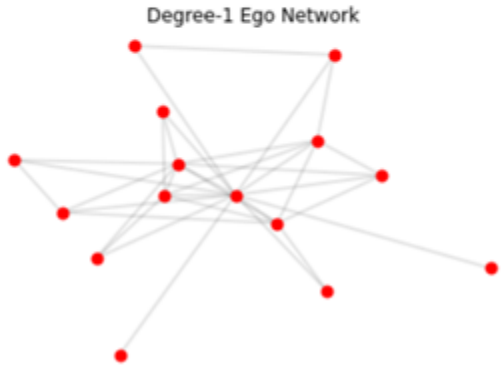
1. Read the data from the amazon-books.txt and populate amazonProducts nested dictionary, key = ASIN; value = MetaData associated with ASIN
2. Read the data from amazon-books-copurchase.edgelist and assign it to copurchaseGraph weighted Graph; node = ASIN; edge = copurchase, edge weight = category similarity. For example a book with purchasedAsin = '0875421210'
3. Get metadata linked to the book.
4. Get the depth-1 ego network of purchasedAsin from copurchaseGraph
5. Use the island method on purchasedAsinEgoGraph to only retain edges with Threshold >= 0.5

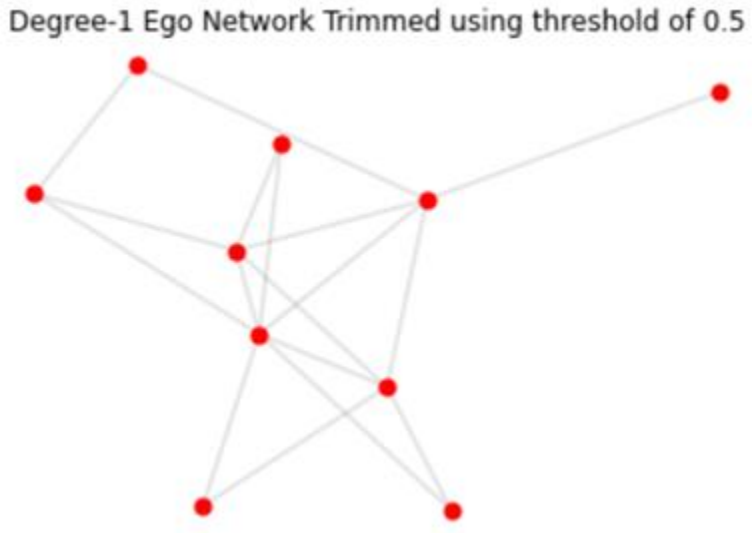
6. Get the list of nodes connected to the purchasedAsin
7. Get Top Five book recommendations from among the purchasedAsinNeighbours based on one or more of the following data of the neighbouring nodes: SalesRank, AvgRating, TotalReviews, DegreeCentrality, and ClusteringCoeff
8. Accessing metadata with ASIN in purchasedAsinNeighbours
9. Print Top 5 Recommendations

Result of Graph based recommendation:

```
Looking for Recommendations for Customer purchasing this Book:

----------------------------------------------------------
ASIN =  0875421210
Title =  Earth Power: Techniques of Natural Magic (Llewellyn's Practical Magick)
SalesRank =  100261
TotalReviews =  47
AvgRating =  4.5
DegreeCentrality =  14
ClusteringCoeff =  0.67
```





```
Top 5 Recommendations by AvgRating then by TotalReviews for Users Purchased the book:

--------------------------------------------------------------------------------
ASIN     Title   SalesRank       TotalReviews    AvgRating       DegreeCentrality        ClusteringCoeff
('0875421229', "Cunningham's Encyclopedia of Magical Herbs (Llewellyn's Sourcebook Series)", 6565, 96, 4.5, 68, 0.65)
('0875421849', "Living Wicca: A Further Guide for the Solitary Practitioner (Llewellyn's Practical Magick)", 6003, 95, 4.5, 30,
0.81)
('0875421318', "Earth, Air, Fire, and Water: More Techniques of Natural Magic (Llewellyn's Practical Magick Series)", 7286, 57,
4.5, 10, 0.73)
('0875421261', "Cunningham's Encyclopedia of Crystal, Gem, and Metal Magic", 14867, 39, 4.0, 17, 0.54)
('0875421245', "The Magical Household: Spells & Rituals for the Home (Llewellyn's Practical Magick Series)", 111836, 21, 4.0,
8, 0.7)
```

Result Discussion of Graph-based Recommendations:

In Ego network set of ties or edges connecting on ego and ego's alters. An Ego is an individual node, in our case it is product id or ASIN to whom you can ask question like category similarity between ASINs. Based on this, we get ego network of purchased ASIN from co-purchased graph. For our implementation of the island method, we made use of the ASIN, to obtain the metadata associated with a particular book. By taking the books that have been co-purchased with this book previously we get the degree-1 ego network. Now we apply the island method on the degree-1 graph, by doing this we can narrow it down to the most similar books. The edges with threshold >= 0.5 are retained. This is how we produced a trimmed graph which contains neighbors of the node with a particular ASIN. We finally get the 5 recommendations for given ASIN.

Code Link : https://drive.google.com/drive/folders/18ys1nISpujRDa4pP4VH_-Yfmgy5Yxy0g?usp=sharing

Conclusion:

Recommendation systems assist consumers in discovering products they may not have discovered on their own and facilitate sales to prospective buyers, providing an efficient method of targeted marketing by providing each customer with a customized shopping experience. In this project we have built a recommendation system based on social network analysis using co-purchased data wherein we have used the Amazon data set for analysis. We propose a graph-based solution based on the Ego network and Island method. Ego network helps to perform efficient learning on graphs and gives information about the individual customer that how his/ her behavior differs with another's. From a micro perspective, ego networks in a complete big network can reveal a lot about its distinction and cohesion. Finally, we say that our improved version of graph-based system for recommendation is an optimized technique for recommender systems which has better efficiency than traditional collaborative filtering methods.

Related Work

The article, "Friends, Strangers, and the Value of Ego Networks for Recommendation" by Amit Sharma, Mevlana Gemici and Dan Cosley discussed two approaches for recommendation with using social networks. One method is augmenting collaborative filtering by using social networks. The second approach is using those algorithms, which only use the egocentric data. Using movie and music data from Facebook and hashtag data from Twitter, the authors have compared the two methods. Even though they need far less data and computational resources, recommendation algorithms based solely on friends perform no worse than those based on the full network. Furthermore, the author's findings indicate that the importance of integrating social network information into recommender algorithms is driven by locality of preference or the non-random distribution of item preferences in a social network. When locality is big, for example as it is in Twitter data then simple algorithms such as K-nearest neighbors recommenders that use only friends perform better than those that use the entire network. These findings

show that systems that see egocentric slices of a full network for example websites that use Facebook logins or have computational limitations for example mobile devices will benefit from using egocentric recommendation algorithms. We studied this article and found out a lot of common reasons how using ego-centric systems can be beneficial. [1]

The article “Ego-net Community Mining Applied to Friend Suggestion” by Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Ismail Oner Sebe, Ahmed Taei, Sunita Verma, the authors have presented a study by investigating the group structure of ego-networks—graphs that reflect relations among a node's neighbors—for several online social networks [2]. To achieve this, the authors have devised a new method for rapidly constructing and clustering all ego-nets graphs. The construction on large networks is very difficult. In the authors experiments the results are compelling high-quality communities that can be detected at a microscopic stage. Authors then use this information to create new features for friend recommendation based on the co-occurrence of two nodes in different ego-network communities. By analyzing the neighborhood of each node, our new features can be computed quickly on very large-scale graphs. Furthermore, the authors have shown that this new similarity test is better than the classic local features used for friend recommendations both formally and experimentally on a stylized model. [2]

“A Structural Approach to Contact Recommendations in Online Social Networks” article by Scott A. Golder, Sarita Yardi, Alice Marwick and Danah Boyd discussed that people tend to form network connections with those people who are like them, however, finding people to communicate with on social networking platforms is not always easy, reducing the network's value for its users. This position paper examines the various ways in which we can make friend recommendations or suggest users to follow on Twitter. The authors have looked at a few different ways to describe similarity, as well as the consequences of these distinctions for community building. For friend recommendations, people can want various things, which has consequences for future research and product design. [3]

The article “Graph based Collaborative Ranking" the authors have discussed the process of item recommendation which is normally complicated by data sparsity and is a common problem in the neighbor-based collaborative filtering domain [4]. This issue is more severe in the collaborative ranking domain, where users' similarities are calculated, and items are recommended based on ranking results. Graph based Collaborative ranking will model users' preferences correctly in a new tripartite graph structure and analyze it to derive a recommendation list directly. As compared to state-of-the-art graph-based recommendation algorithms and other collaborative ranking techniques, the experimental results indicate a substantial improvement in recommendation accuracy. One of the key classes of collaborative filtering is neighbor-based collaborative filtering, which estimates the ranking and rating of the target user based on the behavior of similar users. This, however, is not very effective. This is due to the fact that users only provide feedback on a small percentage of items in recommender systems, and as a result, they rarely have sufficiently common items or pairwise comparisons to estimate their true similarities and dissimilarities. Graph-based recommendation, which takes advantage of heterogeneous information networks, which are information networks with different types of nodes and edges, to refine the similarity measurements, is one optimized approach to overcome this problem.[4]

In the article, “Evolution of Ego-networks in Social Media with Link Recommendations” Luca Maria Aiello and Nicola Barieri, the authors have discussed that even though ego-networks are fundamental structures in social graphs, the mechanism of their evolution remains largely unknown. A key question in the online context is how connection recommender systems can skew the growth of these networks, potentially limiting diversity. To shed light on this, the authors have examined the entire

temporal evolution of 170M ego-networks extracted from Flickr and Tumblr, contrasting links generated randomly with those suggested by an algorithm. The diameter expansion is limited by recommendations that favor common and well-connected nodes. The authors found that the bias introduced by the guidelines fosters global diversity in the process of neighbor selection in a matching experiment aimed at detecting causal relationships from observational results. Finally, the authors demonstrated how findings from our research can be used to enhance the efficacy of social recommender systems using two relation prediction experiments. [5]

References:

[1] “Friends, Strangers, and the Value of Ego Networks for Recommendation” Amit Sharma, Mevlana Gemici and Dan Cosley, 2013 : https://arxiv.org/abs/1304.4837 [1]

[2]“Ego-net Community Mining Applied to Friend Suggestion” Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Ismail Oner Sebe, Ahmed Taei, Sunita Verma,http://www.vldb.org/pvldb/vol9/p324-epasto.pdf [2]

[3]”A Structural Approach to Contact Recommendations in Online Social Networks” Scott A. Golder, Sarita Yardi, Alice Marwick and Danah Boyd https://yardi.people.si.umich.edu/pubs/Yardi_SocialNetworkRecommendations09.pdf [3].

[4] “Graph-based Collaborative Ranking” Bita Shams, Saman Haratizadeh https://www.sciencedirect.com/science/article/abs/pii/S0957417416304912 [4]

[5] “Evolution of Ego-networks in Social Media with Link Recommendations” Luca Maria Aiello, Nicola Barieri 2017 https://dl.acm.org/doi/10.1145/3018661.3018733 [5]