CS224W Project: Recommendation System Models in Product Rating Predictions

Xiaoye Liu

xiaoye@stanford.edu

Abstract

A product recommender system based on product-review information and metadata history was implemented in our project. The primary goal for our recommender system is predicting the rating value that a user will give to a product. We used collaborative filtering model with both user-based and item-based strategies, matrix factorization model and a graph-based Network Inference model as our rating prediction models. We evaluted the performance of these models on Amazon Product co-purchasing Network metadata Dataset[2]. We also discussed the advantages and weakness of them.

1 Overview

1.1 Introduction

Recommendation system has been widely applied to e-commerce and personalized recommending services today, such as recommended friends on Facebook, video recommending on Youtube and music recommendations on Itunes and so on. The benefits that a well-designed recommender system could contribute to business is significant. Since users tend to have higher incentive being interested in the items which satifies their tastes and needs rather than random items, personalized recommendations could help increase sales in retailing and improve the users' experience with services.

Suppose we could predict the numerical rate that a user will give to a product, we could have a better understanding on the preference and taste of this user when making any recommending decisions. The predicted rate could help provide essentially strong evidence to improve the performance of the entire recommending decisions. In our project, we explored several popular rate-prediction models in recommender system and evaluted and compared which achieved highest possible recommendation accuracy.

1.2 Prior Work

There are various approaches and techniques are being developed and applied in recommender system today.In our project, we focused on the similarity strategy and matrix factorization method in Collarborative Filtering and Graph-based Network Inference. Collaborative Filtering (CF) is one of the most popular recommender system strategies. The key idea of similarity strategy in CF is to make inference on a certain subject (i.e: user or item) based on the information provided by its closest neighbors. Thus, the computation of the similarities between subjects based on the existing connections for choosing closest neighbors is the most essential step in CF. In CF, the similarity is only decided based on each subject's preference profile or history, which does not require subject's internal attributes. For example, we decide the similarity of two customers purely based on the items they have purchased before without considering their ages, gender, etc. If we need to decide the similarity between subjects based on their features, such method is named Content-based Filtering. Many other studies also investigate in comparing the performance of CF and Content-based Filtering. We only focused on the analysis of CF in our study.

Matrix factorization strategy is another model in CF which does not require similarity computation to make inference or prediction. Matrix factorization techinique was first successfully introducted into recommender system in Netflix Prize[3] competition. It makes estimation for reponse values via statistical model. If we want to predict the rating value that a user gives to a product, we took the cross product of vector for this user's opinion to all products and a vector of all rating values that this product received. By minimizing the estimation error, we decide the value of latent dimensions the cross product between user and item vector in a latent vector space.

Graph-based Network Inference model is different from CF. The relevance between two users are calculated based on use Bipartite Projection algorithm which uses a graph-based strategy. The predicted rating from a user is determined by the rating from the relevant user to a certain item. However, the relevancy is decided by graph based approach instead of the pairwise similarity we previously introduced.

Specifically, we introduced and evaluated the rate-prediction performance of Collaborative Filtering for user-based, item-based[2], and matrix Factorization strategies, and graph-based Network Inference model. We worked with a large-scale dataset – Amazon product co-purchasing network metadata.[2]

1.3 Datasets

The data was collected by crawling Amazon website including product metadata and review information. There are 548,550 different products. The dataset includes various information for each product and we extract the ASIN, title and review information for each product. There are 7,593,244 unique reviews extracted. From all the review information data, we obtained customerID, rating score. By extracting user information from product review section, we have

1,555,170 unique users extracted, who gave rates and reviews to the 548K products. The grand average for user review rating is about 4.17. Table 1 includes the general information for Amazon dataset.

Type	Number
Products	548,550
Reviews	7,593,244
UserIDs	1,555,170

Table1: Amazon co-purchase Network Dataset Information

2 Methods

1. Collaborative Filtering:

User-based Model: We predicted ratings for a user m to a product p based on the known ratings from m's closest neighbors' ratings given to p. First, we need to decide how we should choose the closest neighbors. We calculate the cosine similarity between the user m and all other users. We started with a matrix where it records each user's review ratings for all items, where each row and column represents per user and per item respectively and each entry is the corresponding rating score. It is very likely that users have never left any rating record for most of the items, thus the distribution for the input matrix is quite sparse. We took the cosine similarity between the vectors of two users. For example, if there is a pair of user m and n, the cosine similarity will be:

$$W_{mn} = \frac{|R(m) \cap R(n)|}{|R(m)||R(n)|}$$

Then, we select the k nearest neighbors of user m based on this similarity measure. Compute a overall predicted rating for product p from user m based on the weighted combination of all m's k nearest neighbors ratings. We also compared how different size of k influence the accuracy in the later evaluation section. During the actual computation and estimation, we notice that the many users did not have any previous ratings and their neighbors also did not have correspond ratings during the prediction. As previously mentioned, it is due to that the data matrix is very sparse. For the purpose of computing, we decided to replace the missing ratings with the grand average score for all review ratings which is 4.17, in order to proceed further analyzation.

Item-based model: Our goal is to find out the rating for item i from a certain user m. The fundamental idea for item-based model is the same as the user-based model. We also applied the cosine similarity function to compute the similarity on an item-item base. For an item i, we first found out its similarity to all other items that user m had rated before and get the rating information for i's k closest neighbors (items). As mentioned previously, sparse matrix leads to challenging computation due to the missing information. Similiary, we also replaced the missing rate to be 4.17 for all the necessary ratings when computing the overall rating from an item's k nearest neighbor.

Matrix factorization: The assumption of matrix factorization is that a similarity layer between users and items is induced by a hidden lower dimensional structure present in the data. We presented each item i with a vector Q_i and each user m with a vector P_m. We then use the resulting dot product $Q_i^T P_m$ to present the relationship between user m and item i. From the result of the dot product we can make a prediction to the rate from user m to item i. During the training process, we used stochastic gradient descent approach to adjust ourQ_i and P_u minimize the prediction error.There are two common approaches mentioned in Netflix competition winning paper [2], alternating least square and stochastic gradient descent. Due to that our dataset has a sparse distribution, we decide to use stochastic gradient descent approach to train our latent factors.

3. Network-based Inference:

The Network-based Inference approach is a graph-based approach. We basically can think about our users and the items are two sets of nodes X and Y. We first need to find out the connection between each user to each item. And from each X_i and Y_i connections, we could get the projections for the set X and Y. In other words, we can get the inner connection relationship within set X and Y.(Figure 1)

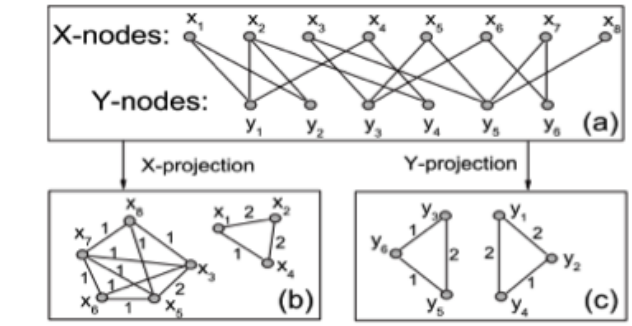

Figure 1: Sets relationships for Network-based Inference Model [5]

Then, it is easy to see that for a given node, either could be user m or item i, which other nodes are connected to it and we treated those connected ones as the given node m's neighbors to make further estimation to review ratings. The new estimated ratings are also computed as a weighted combination of its neighbors' rating scores.

3 Evaluations and Challenges

In our project, we used root-mean-square error(RMSE) to evaluate the performace for each method on the test set. We normalized the RMSE result by dividing the max range of the possible rating scale.

1. Collaborative Filtering:

User-based Model: For user-based model we randomly chose 10,000 users from the total 1.5 Million unique users as our testing set. We perform the user-based CF on the Amazon dataset. The RMSE value results is shown as below with different k size for the nearest neighbors:

size of K	RMSE
k=1	0.3707
k=5	0.3612
k=10	0.3437
k=15	0.3357
k=20	0.3332

Table 2: user-based RMSE result with different size of neighbors k

We are actually expecting the value of RMSE should give a increasing pattern as we increase the number of neighbors. It is because that the more neighbors involved in the estimation computation, the weaker connections we will be imposing to the computation processes. However, surprisingly that our RMSE result is decreasing here. After thinking through the possible reasons, we conclude that the decreasing pattern is due to the highly sparse distribution of the dataset. When computing the user-user similarity, it is very possible that users rated on different items and have less common rated item or the neighbors did not rate on the target item we are interested in.

Item-based Model: In item-based model, we chose the similar items for an item i from a certain set of itemsets which had already been rated by a particular user. This could possibly lead to a better computation accuracy since all similar items are gurantee rated. However, it is also possible that we started with a cold start since a certain could possibly never rated any items. The result for item-based CF is shown as below (Table 3):

size of K	RMSE
k=1	0.2132
k=5	0.2107
k=10	0.2285
k=15	0.2357
k=20	0.2462

Table 3: item-based RMSE result with different size of neighbors k

We can see that the RMSE values gave increasing pattern as we expected. As the value of k increases, it means that more of the similiar items to our item i that a certain user rated before would join in the rating prediction computation. However, although the items are similar, the ratings could be very diverse for each of them since they are all rated by the same user. The user tend to give ratings with larger variation towards similar items. Therefore, smaller size of k will decrease such variation during the computation and vice versa. Thus, we can see that the RMSE result is increasing as the value of k increases.

In addition, the item-item based has a lower RMSE value than the user-user based in general. It indicates that item-based CF is giving slightly better performance than user-based CF on this Amazon dataset.

Matrix Factorization: Due to the limited computation resources, we could not finish the training and the testing for the full amazon dataset for matrix factorization method. Therefore, we decide to reduce the computation dimensions and run the matrix factorization approach on a reduced dataset. For the reduced dataset, we randomly selected 50k users as our training set. All selected users are pulling from the set who gave at least five review ratings. We also chose 10k as our testing set. The 50k users associate with 115,257 different products and 10k users associate with 27,891 products.

When computing over the traing set with different latent factor value l, we are getting the result as below

latent factor	RMSE
l=5	0.2401
l=10	0.2450
l=15	0.2482
l=20	0.2499

Table 3: Matrix factorization RMSE result with different size of latent dimension

Although due to the limited amount of time, we did not perform regularization as mentioned in original Netflix competition paper, we get the result showing that as the value of latent factor increase the RMSE is also increasing. Although we chose reduced dataset, the performace of matrix factorization is better than user-based CF model but slightly worse than the item-based model. If we could apply larger training set for the matrix factorization model, we could also expect a even better performance of it.

2. Network-Based Inference:

The graph-based network inference strategy is actually giving the worse performance on our testing set. This is slightly out of our expectation. The RMSE result is approximately 0.2392 for all 10k testing set users. The RMSE value indicate that it gives better performance than user-based model and the matrix factorization model(for the reduced set). However, item-based model is performing better when the neighbor size is smaller than k=15. It could be due to that although the users within the same set of x are connected, there could still be fairly large variation to their ratings to a certain item. The connected nodes for users or items might not be so strongly connected. The strength of the connection is the key for the prediction accuracy.One challenge we noticed during the computation is that the network-based inference model performs poorly when users do not have enough review data and thus difficult to find the inner set connections.

3.1 Summary

We performed user-based and item-based strategy in CF and also the network-based inference model for the full amazon dataset. From the RMSE result

comparison we can see that item-based strategy gives better performance of all for an input dataset which has a highly sparsed distribution. The matrix factorization was expected to give a btter performance. However, due to the limited computation size and time, we were not able to finish its computation over the full amazon dataset. However we can still see that it gives fairly competitive performace over the reduced dataset.

3.2 Thanks!

Thanks for all the help from the TAs and ProfessorLescovec!

4 References

1. Amazon product co-purchasing network metadata http://snap.stanford.edu/data/amazon-meta.html

2. Collaborative Filtering Recommender Systems J. Ben Schafer, Dan Frankowski, Jon Herlocker, Shilad Sen http://link.springer.com/chapter/10.1007/978-3-540-72079-9_9

3. MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS Yehuda Koren, Robert Bell and Chris Volinsky https://datajobs.com/data-science-repo/Recommender-Systems-

4. Supervised Random Walks: Predicting and Recommending Links in Social Networks Lars Backstrom, Jure Leskovec http://snap.stanford.edu/class/cs224w-readings/backstrom11randomwalk.pdf

5. Bipartite network projection and personal recommendation Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang http://pre.aps.org/pdf/PRE/v76/i4/e046115