Social Relationship Recommender System

Sibirajan Sadhasivam
Department of Computer Engineering
San Jose State University
Student-Id : 013785369
San Jose, CA

***Abstract*—The development of social network has increased the importance of social recommendation. However, social recommender systems have only recently been given research attention. Social relationships between users, especially trust relationships, can facilitate the design of social recommender systems. Such systems are based on the idea that users linked by a social network tend to share similar interests. Existing recommender approaches based on social trust relationships do not fully utilize such relationships and thus have low prediction accuracy or slow convergence speed. I will be proposing a model which be providing relationship advice to the user based the collaborative filtering model. Data set to be used is obtained from user in the form of Google forms.**

Keywords:

I. Introduction

From the establishment of Facebook, WeChat and etc., social network has been an essential part of our life. Web users generate a huge volume of information every day. It becomes increasingly important to find and deliver personalized and useful information to each user. Therefor social recommender systems have been adopted in the social network. Its key idea is to discover relevant information and predict user behavior with data mining and social recommender methods.

Social recommender system has an interdependent relationship with social network. Social network supplies abundant raw user data, i.e. users' profile, comments, tags, friends, to social recommender systems. Social recommender systems provide personalized and precise results to the users, improve the user experience of the social network, and help the social network to attract more users.

The collaborative algorithm is one of the most successful and widely applied social recommender algorithms. It was first proposed by Goldberg, Nichols, Oki and Terry in 1992. They built a system called Tapestry with collaborative method to filter the e-mails. Though the system could only cover a small amount of users and require the attributes of the users, it indeed gave a new recommender method.

Collaborative filtering supposes the users with the same interests may like the similar items. Different from the content-based recommender algorithms, collaborative filtering does not need to identify the content of items but to collect the rating of users to them and gives the recommender results to users finally. It collects the rating of many users in social network and filters the disorder and unrelated information. It has been applied to electronic commerce and personalized online community.

II. The Basic Collaborative Filtering Algorithms

To approach this problem, we had certain factors to be considered. Before investing in a company, venture capitalist should look for the following, Collaborative filtering calculates the similarities of different users and gets the recommendation results accordingly. It picks users/items that have high similarity with the user/item of interest as the neighbors. The characteristics of the neighbors (i.e. ratings, comments, behaviors) will then be evaluated and be used for prediction and recommendation for the user/item of interest.

Two of the main collaborative filtering types are user-based collaborative filtering and item-based collaborative filtering. They all belong to the memory-based collaborative filtering. Another major algorithm is model-based collaborative filtering and we mainly talk about the former algorithm in this paper[13].

1) User-Based Collaborative Filtering: User-based collaborative filtering (UBCF) takes the users with the same rating to a given item as a user set. It then predicts the user's rating to another item according to others' rating in the same user set.

The critical part of the UBCF algorithm is to find the best user set (neighbors) for the chosen user. In other words, the key is to find the neighbors with the greatest similarities with the chosen user. After getting the similarity of the user to others, we choose the similar neighbors of users according to the similarity. At last, we predict the rating of users to specific items with the rating history of neighbors and get the recommender results.

We define the average rating to all items of a user

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{I_j \in I_i} v_{ij}. \tag{1}$$

vij is the user is rating to item j, Ii is the set of items rated by user i. According to the average rating and user similarity, we get the prediction equation

$$p_{i,j} = \bar{v_i} + \kappa \sum_{k=1}^{n} w(i,k)(v_{kj} - \bar{v_k}) \tag{2}$$

There are two key steps to solve this equation. At first collecting the all rating of items of the chosen user. Then calculating the similarity of the users. There are two general methods to calculate the user similarity, one is to calculate the cosine similarity and the other is to calculate the Pearson coefficient. The cosine similarity is defined as

$$w(u,v) = \frac{\sum\limits_{i \in I} r_{u,i} r_{v,i}}{\sqrt{\sum\limits_{i \in I} r_{u,i}^2} \sqrt{\sum\limits_{i \in I} r_{v,i}^2}} \tag{4}$$

Among the equation, ru.i means the rating of user u to item i. To eliminate the limitation of just considering the similarity of the dimension but not the difference of different dimensions, we revise the cosine similarity by subtracting the mean value from every dimension, then we get a revised cosine similarity equation and it is widely applied in calculation of the similarity

$$w(u,v) = \frac{\sum\limits_{i \in I} (r_{u,i} - \bar{r_i})(r_{v,i} - \bar{r_i})}{\sqrt{\sum\limits_{i \in l} (r_{u,i} - \bar{r_i})^2} \sqrt{\sum\limits_{i \in I} (r_{v,i} - \bar{r_i})^2}} \tag{5}$$

The second way of calculating the similarity is to get the Pearson coefficient that is the covariance of the rating of user u and v to corresponding items. The equation is

$$w(u,v) = \frac{\sum\limits_{i \in I} (r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum\limits_{i \in I} (r_u - \bar{r_u})^2} \sqrt{\sum\limits_{i \in I} (r_{v,i} - \bar{r_v})^2}} \tag{6}$$

For the discovery of the nearest neighbors, there are two general methodsk nearest neighbors (KNN) method and setting threshold method.

K nearest neighbors means to choose the k nearest (i.e. the highest similarity) users. If we want to choose the top 3 nearest neighbors of point 1. We set k=3, then choose the top 3 nearest points as the neighbors: 2, 4 and 7.

Setting threshold is to give a threshold at first, if the similarity between user u and v is bigger than , then user v will be chosen as the neighbor, either will not be chosen. The nearest neighbors are in a circle whose center is user u and radius is the threshold. In Figure 2, we set threshold equals to k, so point 2,4,7,3 will be chosen.

Comparing with two methods, KNN will choose the neighbors with the highest similarity but if the value of k is too big, the accuracy will decrease. For the threshold setting method, the number of neighbors may be small in some situation but there will not exist large difference among chosen neighbors. KNN has an advantage that no matter how low the similarity is, there are always k neighbors will be chosen. Therefore, KNN is widely applied in reality.

The last step is to generate the recommender results. We need to predict the rating of user u to a specific item i. The general method is to calculate the average rating of chosen neighbors to item i ru,i is the predicting rating of user u to item I. N is the set of similar neighbors of user u,rvi is the rating of neighbor v to item i.

To improve the method, two algorithms using weighted averaging are proposed. The first algorithm is to calculate the weighted average of near neighbors. The other one is to calculate the increment of users' rating then calculates the weighted averaging of it. Two equations are shown below simu,v represents the similarity between user u and its bear neighbor v. From the equation we could find that the influence of neighbor to user increases in directly proportion to the similarity between them.

2) Item-Based Collaborative Filtering: Item-based collaborative filtering (IBCF) is to compare the similarity of different items, then to predict the rating to a similar item of a user according to its current rating of items. The item can be figured as film or movie in reality.

Like the UBCF algorithm, we need to collect the rating to different items of the same user. The three steps of predicting are the same as the UBCF.

In IBCF we define the prediction of the user i to item j is

$$p_{i,j} = \kappa s \sum_{k=1}^{m} w(k,j) \bullet v_{i,k} \tag{10}$$

K is the normalization factor defined by (3). w(k, j) is the similarity of items and calculated by cosine similarity defined by (4).The reason of discarding Pearson coefficient is that IBCF only collect the rating to items of the same user and ru equals to rv in this situation. KNN is also applied in the item-based collaborative filtering. The difference is that the similar neighbors of item i is chosen among the rated items of user u.The generation of recommender results uses the same algorithm as UBCF.

III. DataSet Overview

The dataset for the project was collected using google forms. The google form consisted of 10 questions for which the user will be providing ratings which can vary between 1 and 10. The link for the Google form is https://forms.gle/RqxwQ3L3pexpy3KN8. Fig.1 shows the Google Forms screenshot which was circulated and Fig.2 shows the Dataset screenshot in the form of a csv file.

IV. Data Visualization

The next step in preprocessing is data visualization. "A picture is worth a thousand words" and once the dataset is refined, it is exposed to visualize the patterns, trends and correlations. Our data was visualised to find the relationship between certain features for a better understanding of the dataset before building a model.

Fig.4 and Fig.5 are the histogram plots for the first 4 questions that shows the distribution of the ratings by different users

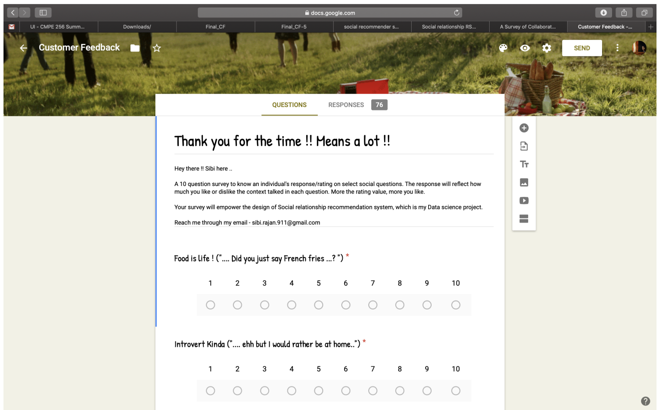

Fig. 1. Google Form Screenshot

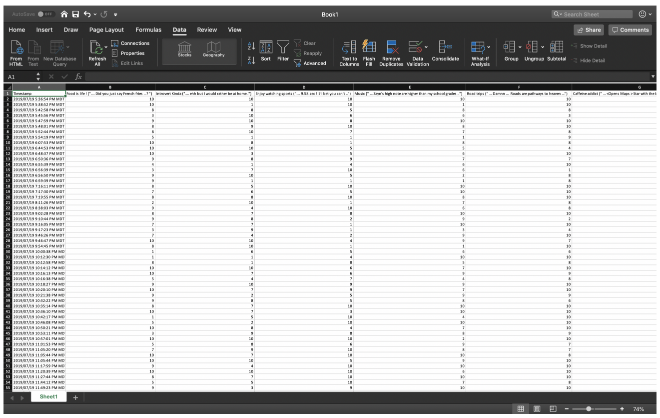

Fig. 2. Dataset screenshot

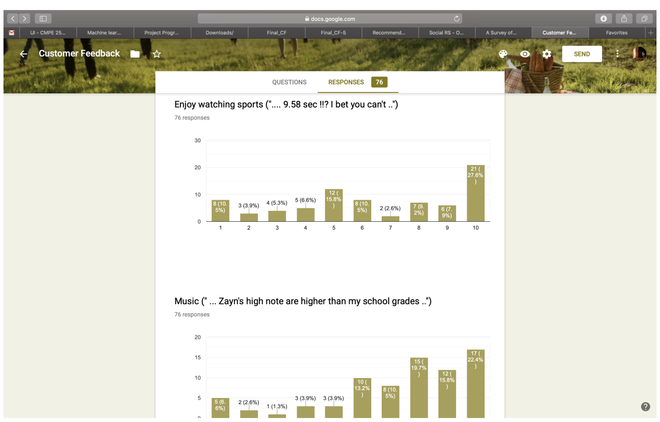

Fig. 3. Question 1 and 2

V. Recommendation model Results

Precision and Recall are recognized as evaluating indicator about the recommendation effect of the recommended system. Formula for precision and recall shown in fig6

where, R(u) is the list of recommendations based on the user's behavior in training set and T(u) is the user's behavior in testing set. Although Precision and Recall is not necessarily related in the calculation formula, it is unlikely to achieve high Precision and high Recall at the same time in an actual recommendation system. Billsus and Pazzani proposed the F indicator to get an equilibrium point between Precision and Recall to evaluate the recommendation system. F indicator can be calculated with the formula shown in fig7

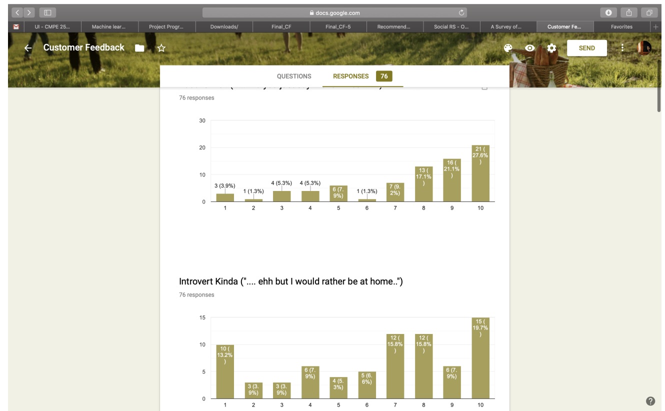

Fig. 4. Question 3 and 4

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Fig. 5. Precision Recall formula

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Fig. 6. F1 indicator formula

All experimentation ran on Windows8.1 with Python3.

VI. Results Analysis

According to the test results, conclusions can be drawn as follow:

- When the number of recommended users is the same and the number of neighbors is similar, using the Jaccard similarity can get better results
- When the recommended number of users is the same and using the same similarity calculation rule, the recommended effect is showing an upward trend with the number of neighbors increasing. When the number of neighbors run up to about 100, the recommended effect achieves the best effect. After that, the effect declines slowly and then tends to stabilize
- For three different users, recommending category, recommending 5 users can get the highest Precision but the lowest Recall. Recommending 10 companies will get the medium effect. Recommending 15 companies will get the highest Recall and best total effect (the highest F value).

VII. Conclusion

I have successfully implemented User based Collaborative filtering and Content based filtering in this paper. This helps in recommending the right list of users and providing relationship advice. Matching users with mutual interest in each other is as important task for online dating sites. In this paper, I used a set of similarity based recommendation algorithm for online dating, that characterises the attractiveness and interest between two users, and select most compatible users for recommendations. The results show that the collaborative filtering based algorithm achieve much better performance than content-based algorithm in both precision and recall and both significantly outperform previously proposed approaches.

VIII. Acknowledgment

I would like to thank Professor Shih Yu Chang and our Teaching Assistant Mr. Surya Sonti for guidance and support in successfully executing this project. I would also like to thank our team members who worked long hours and on weekends to make this happen.

References

[1] Yingtong Dou, Hao Yang,Xiaolong Deng;A Survey of Collaborative Filtering Algorithms for Social Recommender Systems