ISSN:2231-0711

Manusmriti/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 243-246

Implementation of Social Hybrid Product Recommender

Manusmriti
Research Scholar
Department of Computer Science and Engineering
MDU University, Haryana, India

Abstract: Recommender systems represent user preferences for the purpose of suggesting items to purchase or examine. They have become fundamental applications in e-commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences. This paper surveys the landscape of actual and possible hybrid recommenders, and introduces Social hybrid product recommender, a system which combines multiple similarity matrices derived from heterogeneous implicit (User-item rating network) and explicit social networks (Friends network).

Keywords: Friends network, user-item rating network, recommender systems, e-commerce

1. INTRODUCTION

Recommender systems[1,2] were originally defined as ones in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients" (Resnick & Varian 1997[3]). The term now has a broader connotation, describing any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options. Such systems have an obvious appeal in an environment where the amount of on-line information vastly outstrips any individual's capability to survey it. One common thread in recommender systems research is the need to combine recommendation techniques to achieve peak performance. All of the known recommendation techniques have strengths and weaknesses, and many researchers have chosen to combine techniques in different ways. That is why we use hybrid recommender system[4] which combines the strength of both collaborative and content-based systems. This article surveys how we combine multiple similarity matrices derived from heterogeneous implicit networks and explicit social networks.

2. PROBLEM STATEMENT

Often most recommender systems operate in two dimensional User Item space i.e. those users (peer) are identified that has similar rating preferences for items to the current (active) user. Active user who have not seen the product (say S) yet, prediction of that product S for the active user is calculated on the basis of rating of the same product S given by the peer users of the active user. These recommender systems give their recommendation only on the information based on the user item ratings, in other words based on user and item information and ignore to take additional information or do not take into consideration additional information that may be important for the recommendation. In many situations the importance of a certain product to a user may depends on time (product which seems important in present time may not be important in the future for the user), also depends product will be consumed under which circumstances. In such situations simply recommendation of item to users is not worthy, a good recommender system must take additional relevant information of the active user for example place, age and company etc of user into consideration while recommending a product.
Social influence plays very important role on everybody, for recommendation we must understand the role of social influence on one's life. Other people can affect our emotions, opinions and behaviors is the effect of social influence. Project leaders, family member, friend's groups, social class and culture are the social factors which influence our opinions. People have many roles such as wife, mother, employer, employee etc. and these roles change continually.

3. PROPOSED WORK

A hybrid recommender system which combines multiple similarity matrices derived from User-item rating matrix and also from Friends network by checking rating of your peer in the network and use Cosine similarity to compute similarity among users of a bipartite graph and weighted sum method to compute the similarity among users of a uni-partite graph. User-user similarity taken from friends network is considered to give the influence of social factors and leads to better result since our proposed system will give recommendations in user-centric/ ego network means by making one node ego and considering neighborhoods as alters we calculate similarity.
Our system will use a string similarity matching algorithm to match the product keywords with the keywords retrieved from the user search history for a product. The string similarity algorithm will be used to reorder the list of recommended products, with those products first which are being searched by the user in the past. For each target user our system calibrates the influence of each social network. For example, if a user have very few friends in the friend's network and have rated many items in the user-item rating network then the weighting strategy of our system promoted the information given by the user-item rating network. Our proposed system will be fast and free from common limitations of traditional recommender systems.

4. System Implementation

Rating Prediction Based On User-Item Rating Network

As described earlier, social hybrid recommender system will use implicit social network (user-item rating network) to form a user-item similarity matrix.

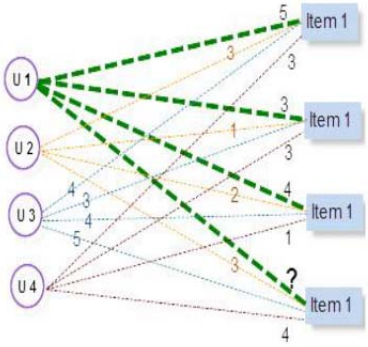

Figure 4.1 User-item rating network

The above shown bipartite graph which is also shown by a matrix R below, where R(*u*,*i*) is the rating of a user *u* over an item *i*.

	Item 1	Item 2	Item 3	Item 4
U1	5	3	4	?
U2	3	1	2	3
U3	4	3	4	5
U4	3	3	1	4

Table 4.1 User-item rating matrix R

Table 4.1 represents rating matrix, where U1-U4 are users and here possible rating values are vary and defined on a numerical scale from 1 means strongly dislike to 5 means strongly like. The cell with no rating is represented by a question mark. Now to calculate the rating similarity matrix SimR, we will use collaborative filtering approach. The idea behind collaborative filtering is simply as follows: given a user-item rating database and the ID of the current user as an input, identify other peer users that had similar rating preferences to the current user in the past. A rating prediction is computed for every item that the current user has not seen yet. The ratings computed for an item depends on the ratings given by the peer users for that item. The underlying assumptions of such methods are that

(a) if users had similar tastes in the past they will have similar tastes in the future

(b) user preferences remain stable and consistent over time.

Related work in Collaborative Filtering has used Cosine similarity or Pearson correlation to calculate the similarity among users (simR). here cosine similarity is used (Equation 1) to measure the similarity between two users u and v, where ru,i= R(u,i).

$$sim(u,v) = \frac{\sum_{\forall i \in I}(r_{u,i} * r_{v,i})}{\sqrt{\sum_{\forall i \in I}(r_{u,i})^2}\sqrt{\sum_{\forall i \in I}(r_{v,i})^2} \qquad (4.1)$$

Using equation 1, the rating similarity matrix SimR can be computed as

	U1	U2	U3	U4
U1	1	0.766	0.783	0.669
U2	0.766	1	0.975	0.916
U3	0.783	0.975	1	0.936
U4	0.669	0.916	0.936	1

Table 4.2 user rating similarity matrix simR

We have computed the user rating similarity matrix simR using equation 4.1. Now to predict the rating of user 1 on Item 4 use equation 2, which is as follows:

$$pred\ rating_{u,i} = \frac{\sum_{v \in U}[sim(u,v) * r_{v,i}]}{\sum_{v \in U} sim(u,v)} \qquad (4.2)$$

Using equation 4.2, the rating prediction of U1 on item 4, is equal to 4.007 [((0.766*3)+(0.783*5)+(0.669*4))/(0.766 + 0.783 + 0.669)].

Equation 4.1 has one drawback that it does not consider the fact that users are different with respect to how they interpret the rating scale as one user may rate same item with high rating number and other may rate with low scale. Some users tend to give only low ratings, whereas others will never give a 5 to any item. To deal with this equation 4.2 can be modified as:

$$\text{pred rating}_{u,i} = avg + \frac{\sum_{v \in U}[sim(u,v) * |(r_{v,i} - avg_v)|]}{\sum_{v \in U} sim(u,v)} \qquad (4.3)$$

Where *i* is any unrated item, avg is average rating of items by user *u* in rating matrix R. Also avgv is average rating of items by user *v*. Corresponding terms in the previous summations of equation 3 should be deleted if some user *v* has not rated item *i*. In the above example we have considered the similarity value of every user to calculate the rating prediction of user 1 on item 4. But in real situation there will be many users (neighbors), so instead of considering similarities of all users, take into account the similarities of say top n users.

Rating Prediction Based on Friend's Network

A Friend's network is also known as peer to peer influential network because each user is influenced by the actions of its peer users (neighbors). Rating prediction using friends network assumes that any item which your closest friends rate will be a good recommendation for you. Friends network is subject to the principle of homophily, which states that a person has relationships with people who are most like him and therefore the person's preferences and his friend's preferences will be highly correlated and a person's preferences are highly motivated by those around him- so if a friend of mine like a product then I am more likely to try it. Using Friend's network and calculate a user-user similarity matrix, SimS and using the same approach (equation 2) discussed in the previous section and calculate the rating for unrated items.

Let G be a simple unipartite graph where there are no multiple edges between the nodes and there are no self loops on any node. The graph is undirected and the edges are weighted according to the relationships between the nodes. The weights represent the closeness (similarity) between the nodes

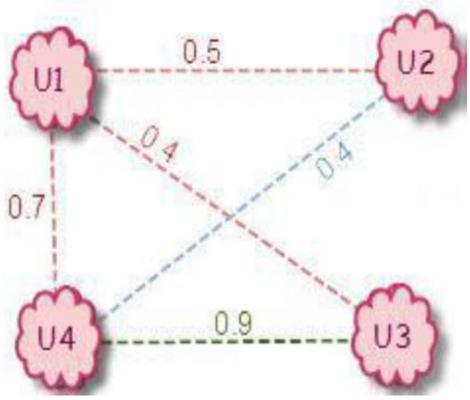

Figure 4.2 Shows Unipartite Friend's Network

There is a variety of similarity measures that can be used to compute the user-user similarity matrix like Random Walk with Restart, FriendTNS, Common Neighbors index, Jaccard Coefficient, Adamic and Adar index etc. for analyzing the closeness of nodes in a network.

The adjacency matrix for the above unipartite friend's graph can be shown as

	U1	U2	U3	U4
U1	0	1	1	1
U2	1	0	0	1
U3	1	0	0	1
U4	1	1	1	0

Table 4.3 shows user-user adjacency matrix S

	U1	U2	U3	U4
U1	0	1	1	1
U2	1	0	0	1
U3	1	0	0	1
U4	1	1	1	0

Table 4.4 shows user-user similarity matrix simS

By considering the above user-user similarity matrix and equation 2, the rating prediction of U1 on item 4, is equal to 3.937 [((0.5 * 3) + (0.4 * 5)+ (0.7 * 4))/(0.5 + 0.4 + 0.7)].

Unifying simR and simS Into A Single Matrix

Combine both the similarity matrices into a single matrix by using equation 4.4 which is as follows:

$$sim(u,v) = \alpha * sim_R + (1-\alpha) * sim_S \qquad (4.4)$$

Calculate similarity between users u and v using equation 4.4, where takes value between [0,1].

Before unifying both the matrices into a single matrix we need to consider the fact that the distribution of similarity values in the interval [0,1] between simR and simS differ significantly. For example the similarity values in simR are distributed between 0.6 and 1, whereas the similarity values in simS are distributed between 0 and 0.9. So, it is unfair to take a simple weighted average of both the similarity matrices using equation 4.4. So first transform both the matrices using equation 4.5.

$$simq(u,v) = \frac{simq(u,v) - m}{s} \qquad (4.5)$$

Where m is the mean similarity of matrix q and s is the standard deviation of matrix q. We then normalize the similarity values in the interval [0,1].

Symeonidis *et. al.* proposed that parameter can be auto adjusted using equation 4.6[5].

$$\alpha = \frac{dR}{dS + dR} \qquad (4.6)$$

where dS= (local S/global S) is the local to global coefficient of the selected user into the user-user adjacency matrix S and dR= (local R/global R) is the local to global density coefficient of the selected user into the user-item rating matrix R. local S is the number of non-zero values in selected user row (adjacency matrix S) divided by the number of users. global S is the number of non-zero values in the full adjacency matrix S divided by the square of number of users. local R is the number of non-zero values in selected user row (user-item rating matrix R) divided by the number of items. global R is the number of non-zero values in the full user-item rating matrix R divided by the number of users and number of items.

Hybrid Product Recommender and Rating Prediction Algorithm

Algorithm: Hybrid Product Recommender and Rating Prediction Algorithm

Input:

U: user for which the item ratings are to be predicted

NUi: where i=1 to k be the immediate neighbors of user u along with their rating on various items.

K: set of keywords for each individual items

H: product search history

Output:

O: set of recommended products

for i= 1 to k

Compute the user-item similarity matrix simR using equation 4.1.

Compute the user-user similarity matrix simS using weighted sum formula.

end for

Transform both the similarity matrices simR and simS using equation 4.5 and then scale them in the interval [0,1].

Compute the local and global densities i.e., local S, global S, local R and global R.

Calculate adjustment parameter □□ using equation 4.6.

Unify both the similarity matrices simR and simS into a single similarity matrix using equation 4.4.

Find top-n similar users U1……….Un to U.

Get the corresponding similarity values of top-n similar users.

Compute the predicted ratings of user U for each unrated item using equation 4.3.

Find the top-e items (items with highest predicted rating) for user U and put them in O.

if there is a product-search history corresponding to user U.

then

Retrieve keywords from search history.

Using Smith Waterman String Similarity coefficient calculate the similarity between the retrieved keywords and the keywords of each top-e items individually.

Reorder the top-e items in the decreasing order of their keyword similarity measure.

Put top-e items in O.

end if

if the predicted items are j less than the required items to be recommended **then**

Add in O, j no. of those items which are recently added to the database (store).

end if return O

5. EXPERIMENTAL EVALUATION

This chapter presents how the recommendations produced by our "Improved Technique for product recommendation and rating prediction" are evaluated. Firstly, the experimental method and the analysis of the dataset (user-item rating dataset, item-keyword dataset, user-user relationship dataset and user-input dataset) are presented. Then compare experimentally various string similarity methods and decide which similarity method would be useful in increasing the accuracy of our social hybrid recommender system. Then show the relationship between top-e items (to be predicted) and the average item prediction accuracy of our recommender system, relationship between top-e items and average keyword-hit accuracy and after that show the relationship between Smith Waterman string similarity coefficient threshold[6] and average keyword-hit accuracy. All our experiments were performed on a 2.3 GHz intel i7 processor with 4 GB of RAM. Java is used to implement all algorithms.

The objectives of the experiments are to verify the effectiveness of our proposed improved technique for product recommendation and rating prediction. To achieve this, the experiments are conducted based on the following hypotheses.

Hypothesis 1: The average item prediction accuracy of the recommenderincreases as the number of top-e items increased.

Hypothesis 2: The keyword-hit accuracy increases as we increase the number of top-e items.

Here top-e items are number of items to be recommended to user.

6. CONCLUSION AND FUTURE SCOPE

In this report, presented a new social hybrid algorithm to make recommendations in the online environment. This report discusses the likelihood of converting social data into quantitative information and using this information to power social recommendations. Our social hybrid product recommender algorithm unifies the similarity matrices obtained from both user-item rating network and friend's network. This social hybrid product recommender can deal with data sparsity problems and cold start problem, and works even when explicit trust rating data in not available. Testing the effectiveness of our proposed recommender system against a large real life social network dataset to verify its suitability in social commerce environment. The proposed algorithm can be redesigned to improve processing time to perform more efficiently, thus not only providing correct recommendations but also faster processing.

REFERENCES

1. http://josquin.cti.depaul.edu/~rburke/pubs/burke-umuai02.pdf
2. https://www.gipp.com/wp-content/papercite-data/pdf/gipp09.pdf
3. Resnick, P. and Varian, H. R.: 1997, 'Recommender Systems'. *Communications of the ACM*, **40** (3), 56-58.
4. Morton, W. 2013. Hybrid Recommender System. (April 2013). Retrieved February 6, 2014 from http://blog.soton.ac.uk/comp6051-bazinga/files/2013/04/HybridRS.png
5. P. Symeonidis, E. Tiakas, and Y. Manolopoulos, "Product Recommendation and Rating Prediction based on Multi-modal Social Networks",(Thessaloniki, Greece),pp. 23-27,2011.
6. https://scholar.google.co.in/scholar?q=smith+waterman+string+similarity+coefficient&btnG=&hl=en&as_sdt=0%2C5&as_vis=1