2011 International Conference on Image Information Processing (ICIIP 2011)

Recommendation Based on Prominent Items

Raminder Jeet Singh Sodhi
Department of Computer Engineering
National Institute of Technology Karnataka
Surathkal , India
rjssodhi@gmail.com

Vaibhav Gaur
Department of Electronics and Communication
Delhi Technological University
New Delhi, India
vaibhav.gaur1990@gmail.com

V.K. Panchal
Scientist'G'
Defence Terrain Research Lab
DRDO, Delhi , India
vkpans@gmail.com

***Abstract*— Today Recommender Systems are used in numerous fields like E-Commerce and Web personalization. Recommender systems support users in the identification of fascinating products, services and people in circumstances where the amount and intricacy of offers exceeds the capability of a user to survey it and to reach a decision.**
Earlier, user-based collaborative filtering (CF) and item-based CF approaches have been used to build recommendation systems but they haven't been able to solve some fundamental problems associated with recommendation systems.
We propose an improved approach of integrating the concept of Prominent Items with nearest 'k' neighbours to solve these problems and make the recommender system effective. We venture into item-based collaborative filtering and classify items as Prominent Items on the basis of which we present a better system of recommendations through intelligently formed user models. Our experimental results show that this approach can achieve better prediction accuracies than traditional item-based CF algorithm and alleviate the problems effectively.

Keywords—**Recommendation, collaborative filtering, extensions to recommender systems, data mining, influence, item-based collaborative filtering.**

I. INTRODUCTION

As a result of whopping information available on web and its unprecedented growth day by day, it has become imperative to develop new technologies to select the useful information and discard the others. Hence the concept of recommendation systems has gained popularity over the years to help users view information according to their desires.
One of the most popular such technologies is *Collaborative Filtering* (CF). The task of collaborative filtering is to predict the rating of a particular user for a particular, yet to be rated or visited, item; or to recommend the top *N* items in which the user may be interested in, based on a database of user votes for the products (or items) on the site. The assumption behind this model is that the active user will prefer those products (or items) that other like-minded users prefer.
There are two CF techniques available today namely user-based CF and item based CF. Though these approaches have been able to produce respectable results, but they still lack in solving the fundamental problems**[1]** associated with Recommender Systems: sparsity, new item problem and fake profile problem.

Firstly, as the number of items in the database increases, the density of each user record with respect to these items will decrease. If the number of users who have rated items is relatively small compared to the number of items in the database, it is likely that there won't be significant similarity between users. Then the user-based CF algorithm will produce less reliable recommendations. That is the sparsity problem of user-based CF approach.

Secondly an item can be recommended only if any user has rated it. In case of new items, as there is no previous rating available for it in the dataset, so it is not possible to recommend it to any user. This is the new item problem.

Finally, the creation of fake user profile to purposely give biased ratings to some items can be detrimental in the process of effective recommendation. This is the fake profile problem.

Though extensive research is going on in this field to solve these problems, but this technique requires an approach which can maximize the closeness of the recommendations to the user's interests. This approach of ours integrate the concept of most influential items with the "k" nearest neighbors concept to devise a new and improved approach called Recommendation based on Prominent Items(RPI).

Our paper is divided 5 sections after the introductions. First section deals with concept of item based CFs, second section deals with our new model RPI approach. The sections afterward deal with the solution mode , experimental results and conclusion.

II. ITEM BASED COLLABORATIVE FILTERING

In a collaborative filtering (CF) scenario, generally we start with a list of m users:

$$U = \{ u_1, u_2, u_3 \cdots u_m \}$$

978-1-61284-861-7/11/$26.00 ©2011 IEEE

and a list of n items :

$$I = \{ i_1, i_2, i_3 \cdots i_n \}$$

and a mapping between user-item pairs and a set of weights. The latter mapping can be represented as a matrix R . In the traditional CF domain the matrix R **[2]** usually represents user ratings of items, thus the entry $R_{i,j}$ represents a user u_i's rating on item i_j . In this case, the users' judgments or preferences are explicitly given by matrix R .

A. Similarity

For the evaluation of direct similarity among users, i.e. $sim(i, j)$, many approaches exist, some of the popular ones being Euclidean Distance, Vector/Cosine similarity, Pearson Correlation Coefficient. These compute the similarity value between each pair of users who have rated a few set of common items. Here, we make use of the standard version of Pearson Correlation Coefficient, which is :

$$sim(i,j) = \frac{\sum_{u \in U} \left(R_{u,i} \times R_{u,j}\right)}{\sqrt{\sum_{u \in U} \left(R_{u,i}\right)^2} \times \sqrt{\sum_{u \in U} \left(R_{u,j}\right)^2}} \qquad (1)$$

B. Recommendation

The significant step of item-based collaborative filtering recommendation approaches is to make quality predictions for target item I_j . Item-based CF approaches use the above defined similarity measures as the distance formulas of k nearest neighbour algorithm and perform kNN query to find the k most similar items for computing the probable ratings $R_{i,j}$ for an active user u_i

$$R_{u,i} = \frac{\sum_{j=1}^{k} \left(R_{u,j} \times sim(i,j)\right)}{\sum_{j=1}^{k} sim(i,j)} \qquad (2)$$

III. RECOMMENDATION BASED ON PROMINENT ITEMS

A. Similarity

The first step includes determining the similarity between the items already rated by some of the users. This can be done by following any one of the following methods: Standard cosine similarity, Adjusted cosine Similarity and Pearson correlation coefficient.

In this paper we propose to use the Pearson correlation coefficient method which is defined in (1).

B. Finding Neighbour

In this step we compute the "k" nearest items for each item in the dataset. This is done by selecting the top "k" items having the highest values of similarity with the particular item**[3]**.

$$N_i = \{Nb_1, Nb_2, Nb_3 \cdots\cdots, Nb_k\}$$

Here N_i represents the set containing the the k nearest neighbours of item i . Nb represents a neighbour.

C. Computing Prominent Items

The computation of the Prominent Items (PI) set is done by the following three steps.

1) Similarity Weighing: Initially, a mean similarity rating is calculated for each item taking into consideration its similarity with other items, the number of users who have co-rated that item and the maximum number of users who have co-rated any item.

$$S_i = \sum \{sim(i,j) * co-rated(i,j) \div \max co-rated\}$$

2) Inverse K: Now, corresponding to each item, a set of items is calculated which have the item under consideration (j) in the set N_i where $i \in I - \{j\}$. This is referred to as inverse "k" set (Ik).

3) Impact value: Then the influence of each item is calculated. The impact rating (denoted by I_i) is defined as the degree to which an item rating match that of the average item rating.

$$I_i = S_i - \left(\sum \left|S_i - S_j\right|\right) / m$$

The top "n" items with the highest I values are considered for the PI set.

D. Recommendation

Recommendation to a particular user u is done on the basis of co-efficient α . For a particular user the value of α is defined as:

$$\alpha_u = \frac{u_{rated}}{\text{Total items}}$$

Where u_{rated} is the number of items rated by a user u .The recommendation can now be made by selecting the top value of $R_{u,t}$ where t is an item not rated by the user.

$$R_{u,t} = \alpha \times \left(\frac{\sum_{i \in N} (R_{u,i} \times sim(i,t))}{\sum_{i \in N} sim(i,t)} \right) + (1-\alpha) \times \left(\frac{\sum_{j \in PI} (R_{u,j} \times sim(j,t))}{\sum_{j \in PI} sim(j,t)} \right)$$

Where N represents the set of "k" nearest neighbours and PI represents the set of Prominent items.

IV. Experimental Results and Performance Study

A. Algorithm Development and Result interpretation environment

The algorithm design of the Recommendation based on Prominent items (RPI) model was carried out on the basis of the theoretical concept proposed above. Subsequently, the algorithm was converted into python code on a core 2 duo @ CPU 1.8 GHz running Linux Ubuntu 11.04, with 2GB of main memory and 160GB of disk platform. The user-item ratings matrix designed in MS Excel was taken from the site, jester.com and used as an input to the Mysql server for generation of user-item pairs. The result sets retrieved from the algorithm's output were ported into Excel sheets through database connection using Mysql. The graphical results are then generated from this data for easy interpretation and efficient analysing of modelled data. Various figures/graphs are introduced to conclude the claim of the hence modelled concept.

B. Dataset

In our experiments we have used the publicly available jester dataset**[5]** which contains 4.1 Million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users. from the jester page (http://goldberg.berkeley.edu/jester-data)[5]. Each user in this dataset have rated at least 20 items. Ratings are real values ranging from -10.00 to +10.00 which are converted to integer values for our dataset. In our experiment, we divide the dataset into 2 parts , a training set and a test set. For the test dataset we use 4% of the parent dataset and calculate the probable rating for the items in this dataset using results calculated from the training dataset.

C. Evaluation Metrics

Several types of measures for evaluating the quality of a recommendation system **[2]** are available. In our paper, Mean Absolute Error (MAE) is adapted as the accuracy metrics by comparing the numerical predictions and the actual user ratings. MAE is a measure of the deviation of recommendations from their true user-specified ratings.

$$MAE = \frac{\sum_{t=1}^{N} |P_t - R_t|}{N}$$

Where P_t represents the probable ratings and R_t represents the actual ratings.

D. Effect of α on MAE

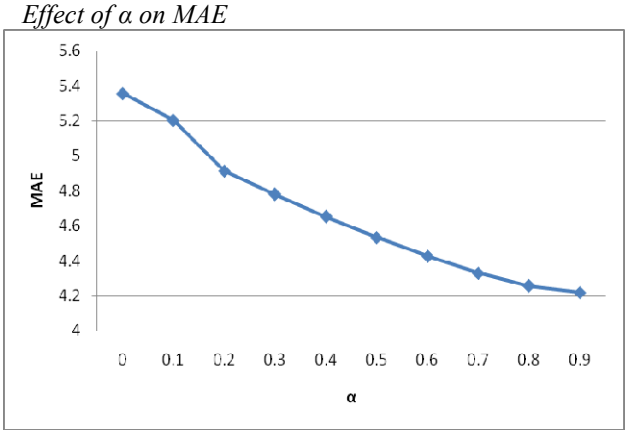

Figure 1: Effect of α on MAE

We can infer from the graph that as the value of α is increasing (i.e. the effect of Prominent item is decreasing on the $R_{u,i}$) the value of MAE is decreasing. This inference can be explained by the fact that we are using a heavily rated system for which the impact of the prominent items should be less.

E. Effect of Neighbour on MAE

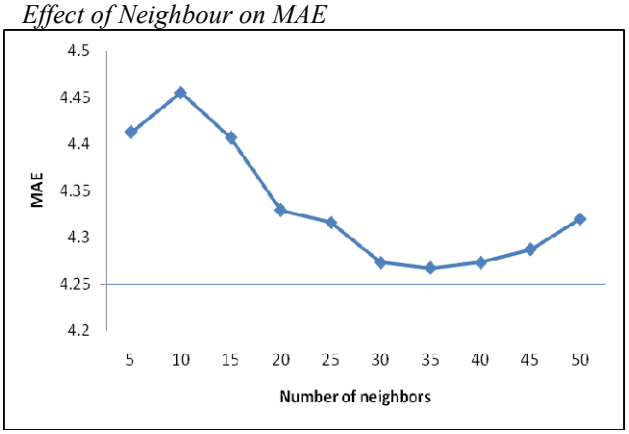

Figure 2:Effect of Neighbour on MAE

This graph has been plotted using α=1 (i.e. the effect of Prominent items has been neglected) .From the graph we can see that the minimum is occurring for Number of Neighbours = 35. We will hence forth use this value for plotting of the graph between MAE and Prominent Items .

F. Effect of Prominent items on MAE

When the number of Prominent items is 3-7 % we are getting a local minima for MAE .There is no benefit of increasing MAE beyond 15% as it would lead to the duplicity of items in the "k" nearest neighbour set and the prominent items set which explains the low value of MAE for large number of prominent items.

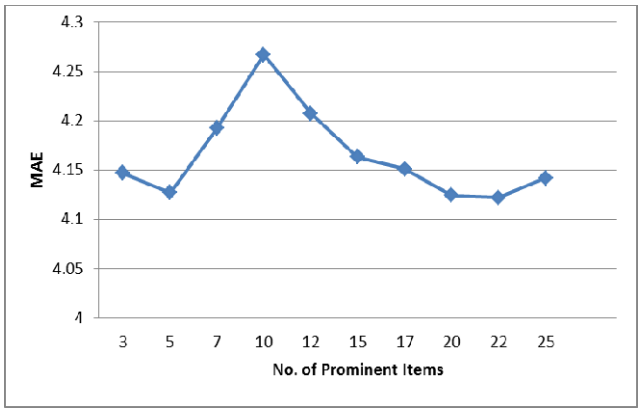

Figure 3:Effect of Prominent Items on MAE

V. SOLUTION MODELS

A. New User Model

In general, the concept of recommendation for a fresh user entering the system seems obscure as there does not exist any preferences for the user so as to map the preferences . However, the Prominent Items can get the new user started with the most popular items present in the community, which proves to be a better proposition rather than random or most viewed/reviewed selection of items**[7]**. This set of most popular items consists of those that are Prominent Items. So, the new user joining gets the most appreciated item recommendations from the database. Meanwhile, as the user rates the other items, a tailored profile modelling process executes concurrently.

B. Quality-Control Model

The concept of Prominent Items proposed insight into the dynamic nature of the hence formed system. Since, the user ratings matrix continuously updates itself; hence the item similarity with others keeps on re-inventing itself. In this self-correcting system, an item are dropped/entered out of the Prominent Items sets and the value of α keeps dynamically changing. Hence, the quality of recommendations is determined by the user community's interest in specific items. This will safeguard the effectiveness of the recommendation as the items in the Prominent Items set will consist only of the most popular items which is determined from the item's ratings given by a community of users and not only just one user(the case of fake profile).

VI. CONCLUSION

Recommender systems are a powerful force for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. They enable users to find item they like and help business by generating more sales. The domain of Recommender Systems has promised to deliver intelligent, reliable recommendations through the extensive research thrusting its efficient adoption to user-centred technologies. The growth of e-commerce market has left the organizations with intimidatingly huge amounts of data to be modelled. Hence the process of finding the Prominent Items helps in modelling the data in the initials stages while maintaining the quality of recommendations as:

A. The prominent items basically represent the most popular items amongst the users which largely affects the recommendation of items to new users.

B. It lends a degree of confidence to the system by limiting the spread of bogus recommendations spawned by the process of unfair bulk ratings through automatic systems .

The projected concept of MIU-based recommendation model has been implemented and verified to stand true to its proposition.

REFERENCES

[1] Analysis of Recommender Systems' Algorithms. By: Emmanouil Vozalis, Konstantinos G. Margaritis. In: The 6th Hellenic European Conference on Computer Mathematics & its Applications (HERCMA), Athens, Greece (2003).

[2] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms.

[3] Recommendation Based on Influence Sets (2006). by Jian Chen, Jian Yin In Proc. of WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis.

[4] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143{177, 2004}

[5] *http://goldberg.berkeley.edu/jester-data/.* The source of the dataset consisting of actual jokes ratings retrieved from the source provider jester jokes.

[6] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM '01: Proceedings* of the tenth international conference on Information *and knowledge management*, pages 247{254, New York, NY, USA, 2001. ACM Press.

[7] *Comparison of Recommender System Algorithms focusing on the New-Item and User-Bias Problem.* By: Stefan Hauger, Karen H. L.Tso2, and Lars Schmidt-Thieme. Book: Data Analysis, Machine Learning and Applications Publisher: Springer Berlin Heidelberg, Pages: 525-532

[8] An MIU [Most Influential Users]-Based Model for Recommender Systems Arora, I., Panchal, V.K.