# The Collaborative Filtering Recommendation Algorithm Based on BP Neural Networks

DanEr CHEN
Zhejiang Textile & Fashion College,
Ningbo 315211, P. R. China
e-mail: chendanerei@sina.com

*Abstract*—**Collaborative filtering is one of the most successful technologies in recommender systems, and widely used in many personalized recommender areas with the development of Internet, such as e-commerce, digital library and so on. The K-nearest neighbor method is a popular way for the collaborative filtering realizations. Its key technique is to find k nearest neighbors for a given user to predict his interests. However, most collaborative filtering algorithms suffer from data sparsity which leads to inaccuracy of recommendation. Aiming at the problem of data sparsity for collaborative filtering, a collaborative filtering algorithm based on BP neural networks is presented. This method uses the BP neural networks to fill the vacant ratings at first, then uses collaborative filtering to form nearest neighborhood, and lastly generates recommendations. The collaborative filtering based on BP neural networks smoothing can produce more accuracy recommendation than the traditional method.**

***Keywords-recommender system; e-commerce; collaborative filtering; BP neural networks***

## I. INTRODUCTION

With the rapid growth and wide application of the Internet, lots of information comes forth to people. However, our experiences and knowledge often do not enough to process the vast amount of usable information. The problem of obtaining needful information becomes more and more serious. To deal with the problem, the personalized recommendation systems play a more important role in many fields [1].

Recommender system plays an important role particularly in an electronic commerce environment as a new marketing strategy. Although a multifarious of recommendation techniques has been developed recently, collaborative filtering (CF) has been known to be the most successful recommendation techniques [2]. The famous electronic commerce website Amazon and CD-Now have employed CF technique to recommend products to customers and it has improved quality and efficiency of their services. The CF assumes that a good way to find a certain user's interesting content is to find other people who have similar interests with him. CF methods operate upon user ratings on observed items making predictions concerning users' interest on unobserved items. The sparsity of ratings problem is particularly important in domains with large or continuously updated list of items as well as a large number of users [3,4,5]. The sparsity problem may occur when either none or few ratings are available for the target user, or for the target item that prediction refers to, or for the entire database in average. Different treatments are required and different prediction techniques must be employed depending on the sparsity conditions, making the selection of an appropriate approach a cumbersome task. Current CF approaches are limited in the sense that they address specific aspects of the above problem.

Aiming at the problem of data sparsity for collaborative filtering, a collaborative filtering algorithm based on BP neural networks is presented. This method uses the BP neural networks to fill the vacant ratings where necessary and uses collaborative filtering to form nearest neighborhood, and then generates recommendations. The collaborative filtering based on BP neural networks smoothing can produce more accuracy recommendation than the traditional method.

## II. USING BP NEURAL NETWORKS TO FILL VACANT RATINGS

### A. BP neural networks

The BP network is based on the supervised procedure and the network constructs a model based on examples of data with known output [6,7,8]. Given a training set { ( x ( t) , y ( t) ) ∣ x ( t) ∈Rm , y ( t) ∈Rn , t = 1 ,2 , ⋯ k} , BP can implement high nonlinear mapping from input to output. There exists a mapping F: Rm →Rn such that F( x ( t) ) = y ( t) .The BP algorithm is carried out as follows:

$$h_j(t) = f(\sum_{i=1}^{m} w_{ji}x_i(t) + a_j),\ j = 1, 2, \dots, p$$

$$y_l^*(t) = f(\sum_{j=1}^{p} v_{lj}h_j(t) + b_l), l = 1, 2, \dots, n$$

where f() is activation function which selected sigmoid function $f ( t) = (1 + e^{-t} )^{-1}$, j and l are the number of neurons of hidden layer and output layer respectively , x( t) = ( x1 ( t) , x2 ( t) , ⋯, xm ( t) ) is input vector , hj( t ) is the output of the jth neuron of hidden layer , $y^*_l( t)$ is the lth output of the network , wji and vlj are weight s between the input/ hidden

978-0-7695-3619-4/09 $25.00 © 2009 IEEE
DOI 10.1109/IUCE.2009.121

layers and hidden/ output layers , respectively , aj and bl are the bias of neurons.

### B. Smoothing vacant ratings

We use a 3-layer neural network as figure 1. For the user-item matrix, we will predict the s user as user.s to the t item as item.t.

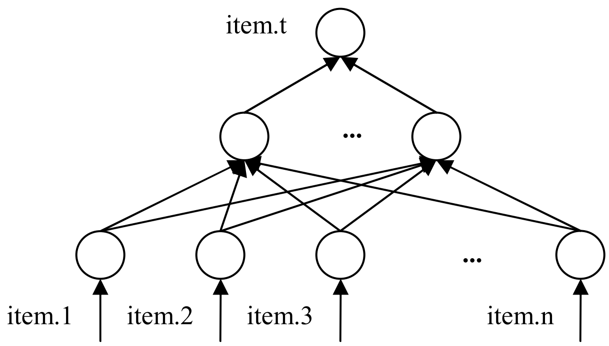


Fig.1 Structure of the BP neural network used to smooth

Let yl ( t) be expected output of neural network. There is an error between actual output and expected output, this error, named mean square error (MSE), can be expressed by the function:

$$E = \frac{1}{2} \sum_{t=1}^{k} \sum_{l=1}^{n} (y_l(t) - y_l^*(t))^2$$

BP algorithm is a gradient descent algorithm, in which the network weights are moved along the negative of the gradient of the MSE function. Input vectors and the corresponding target vectors are used to train the network repeatedly until the error reaches the satisfaction. The trained BP neural network have the ability to generalize, that is, once trained, the system is able to process previously unseen data sample and to yield a probable response.

## III. PRODUCING RECOMMENDATIONS

Through the calculating the vacant user's rating by BP neural networks, we gained the dense users' ratings. Then, to generate prediction of a user's rating, we use the user based collaborative filtering algorithms.

### A. The dense user-item matrix

After we used the BP neural networks, we gained the dense ratings of the users to the items. So, the original sparse user-item rating matrix is now becoming the dense user-item matrix.

### B. Measuring the user rating similarity

There are several similarity algorithms that have been used in the collaborative filtering recommendation algorithm.

Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings.

$$sim(i,j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - A_i)(R_{j,c} - A_j)}{\sqrt{\sum_{c \in I_{ij}} (R_{i,c} - A_i)^2 \sum_{c \in I_{ij}} (R_{j,c} - A_j)^2}}$$

Where Ri,c is the rating of the item c by user i, Ai is the average rating of user i for all the co-rated items, and Iij is the items set both rating by user i and user j.

The cosine measure, as following formula, looks at the angle between two vectors of ratings where a smaller angle is regarded as implying greater similarity.

$$sim(i, j) = \frac{\sum_{k=1}^{n} R_{ik} R_{jk}}{\sqrt{\sum_{k=1}^{n} R_{ik}^{2} \sum_{k=1}^{n} R_{jk}^{2}}}$$

Where Ri,k is the rating of the item k by user i and n is the number of items co-rated by both users.

The adjusted cosine, as following formula, is used in some collaborative filtering methods for similarity among users where the difference in each user's use of the rating scale is taken into account.

$$sim(i,j) = \frac{\sum_{c \in I_{ij}} (R_{ic} - A_c)(R_{jc} - A_c)}{\sqrt{\sum_{c \in I_{ij}} (R_{ic} - A_c)^2 * \sum_{c \in I_{ij}} (R_{jc} - A_c)^2}}$$

Where Ri,c is the rating of the item c by user i, Ac is the average rating of user i for all the co-rated items, and Ii,j is the items set both rating by user i and user j.

In this paper, we use the cosine measure to calculate the similarities of users.

### C. Selecting the target user neighbors

Select of the neighbors who will serve as recommenders. We employ the top-n technique in which a predefined number of n-best neighbors selected.

### D. Recommender using user-based CF

Since we have got the membership of user, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target user.

The rating of the target user u to the target item t is as following:

$$P_{ut} = A_u + \frac{\sum_{i=1}^{c} (R_{it} - A_i) * sim(u,i)}{\sum_{i=1}^{c} sim(u,i)}$$

Where $A_u$ is the average rating of the target user u to the items, Rit is the rating of the neighbour user i to the target item t, Am is the average rating of the neighbour user i to the items, sim(u, i) is the similarity of the target user u and the neighbour user i, and c is the number of the neighbours.

## IV. DATASET AND MEASUREMENT

### A. Data set

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota [9,10]. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings and simple demographic information for the users is included. Therefore the lowest level of sparsity for the tests is defined as 1 – 100000/943*1682=0.937.

The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence. We randomly divided 20% of the experiment data set as test data set and the rest were set as training data set.

### B. Performance measurement

The metrics for evaluating the accuracy of a prediction algorithm can be divided into two main categories [10,11]: statistical accuracy metrics and decision-support metrics. Statistical accuracy metrics evaluate the accuracy of a predictor by comparing predicted values with user provided values. Decision-support accuracy measures how well predictions help user select high-quality items. In this paper, we use decision-support accuracy measures.

Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high-quality items from the set of all items. The receiver operating characteristic (ROC) sensitivity is an example of the decision support accuracy metric. The metric indicates how effectively the system can steer users towards highly-rated items and away from low-rated ones. We use ROC-4 measure as the evaluation metric. Assume that p1, p2, p3, ..., pn is the prediction of users' ratings, and the corresponding real ratings data set of users is q1, q2, q3, ..., qn. See the ROC-4 definition as following:

$$ROC\text{-}4 = \frac{\sum_{i=1}^{n} u_i}{\sum_{i=1}^{n} v_i}$$

$$u_i = \begin{cases} 1, & p_i \geq 4 \ \ and \ \ q_i \geq 4 \\ 0, & otherwise \end{cases}$$

$$v_i = \begin{cases} 1, & p_i \geq 4 \\ 0, & otherwise \end{cases}$$

The larger the ROC-4, the more accurate the predictions would be, allowing for better recommendations to be formulated.

## V. CONCLUSIONS

Collaborative filtering is one of the most successful technologies in recommender systems, and widely used in many personalized recommender areas with the development of Internet, such as e-commerce, digital library and so on. The K-nearest neighbor method is a popular way for the collaborative filtering realizations. Its key technique is to find k nearest neighbors for a given user to predict his interests. However, most collaborative filtering algorithms suffer from data sparsity which leads to inaccuracy of recommendation. Aiming at the problem of data sparsity for collaborative filtering, a collaborative filtering algorithm based on BP neural networks is presented. This method uses the BP neural networks to fill the vacant ratings at first, then uses collaborative filtering to form nearest neighborhood, and lastly generates recommendations. The collaborative filtering based on BP neural networks smoothing can produce more accuracy recommendation than the traditional method.

## REFERENCES


[1] Jong-Seok Lee, Chi-Hyuck Jun, Jaewook Lee, Sooyoung Kim, Classification-based collaborative filtering using market basket data, Expert Systems with Applications 29 (2005) 700–704.

[2] Hyung Jun Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37-51.

[3] George Lekakos, George M. Giaglis, Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors, Interacting with Computers 18 (2006) 410–431.

[4] Tae Hyup Roh, Kyong Joo Oh, Ingoo Han, The collaborative filtering recommendation based on SOM cluster-indexing CBR, Expert System with Application 25(2003) 413-423.

[5] Shih, Y.-Y., & Liu, D.-R., Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands, Expert Systems with Applications (2007),

[6] SongJie GONG, HongWu YE, An Item Based Collaborative Filtering Using BP Neural Networks Prediction, IIS 2009, IEEE CS Press.

[7] Zhang Feng, Chang Huiyou,Employing BP Neural Networks to Alleviate the Sparsity Issue in Collaborative Filtering Recommendation Algorithms, Journal of Computer Research and Development, 43 (4) : 667～672 , 2006.

[8] HE Fangguo, Qi Huan, Back propagation neural network based on modified genetic algorithm and its application, Journal of huazhong normal university (Nat. Sci.), Vol. 41 No. 1 Mar. 2007:51-54.

[9] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International World Wide Web Conference. 2001. 285-295.

[10] SongJie Gong, XiaoYan Shi, A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity, ISBIM2008, IEEE Computer Society Press.

[11] Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan, Personalized Service System Based on Hybrid Filtering for Digital Library, Tsinghua Science and Technology, Volume 12, Number 1, February 2007,1-8.