Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I,
IMECS 2013, March 13 - 15, 2013, Hong Kong

Research on Personalized Recommendation in E-commerce Service based on Data Mining

Tao Xu, Jing Tian, Tomohiro Murata

Abstract —**We propose a new hybrid recommendation algorithm to optimization the cold-start problem with Collaborative Filtering (CF). And we use neighborhood-based collaborative filtering algorithm has obtained great favor due to simplicity, justifiability, and stability. However, when faced with large-scale, sparse, or noise affected data, nearest-neighbor collaborative filtering performs not so well, as the calculation of similarity between user or item pairs is costly and the accuracy of similarity can be easily affected by noise and sparsity. We introduce a new model comprising both In the training stage, user-item and film-item relationships in recommender systems, and describe how to use algorithm generates recommendations for cold-start items based on the preference model. Our experiments model provides a relatively efficient and accurate recommendation technique.**

Key words — **Recommender System, Collaborative Filtering, Data Mining, Data Sparsity, Cold-Start**

I. INTRODUCTION

WITH the rapid spread of Internet, human society is going to enter "information overload" era. How can the users quickly find relevant information from the vast multitude of internet data, which has become a popular topic of Internet. However, the tremendous growth in the amount and variety of available information leads to some austere challenges to recommender systems. [1] Such a situation has induced the so-called information overload problem which leads users finding it increasingly difficult to locate the right information at the right time. [2] Therefore, both researchers and consumers focus on providing more accurate individual information in less time to meet the personalized needs. Personalized recommendation has become a desirable requirement under this background.

Cold-start problem is the most serious problem for collaborative filtering that has not been effectively addressed. As we known the fundamental assumption of CF is that CF analyzes rating matrix to recognize commonalities between users on the basis of their historical ratings, and then generates new recommendations based on liked-minded users' preferences. However, the recommender system can't provide effective recommendations for new user or new item because they have not enough ratings available. Fortunately, quite a number of personalized recommendation systems have collected content information about users and items. Inspire by this reality, we make use of the user or item content information to improve the traditional collaborative filtering.

Collaborative filtering (CF) is currently mainstream technologies of recommender systems. But these approaches have several shortcomings: (1) easy to the influence of noise, (2) if data are very sparse then performance decreases a lot, and (3) limited scalability for the large data sets. In this paper, we propose a new CF method, which is expected to solve the shortcomings of traditional community-based CF method.

The important step in traditional neighborhood-based CF methods is matching similar users, but it is difficult task: computing similarities all user pairs. To solve this problem, we carefully studied the process of seeking for neighbors and find that, among all the users some are quite 'popular' as they are often chosen as neighbors of other users.

The paper is organized as follows: Section 2 describes the recommendation affected and its functional modules; Section 3 introduces the training data set based model main idea and training data set algorithm; Section 4 introduces the rating behavior analysis; Section 5 describes the experimental evaluation; Section 6 summarizes our contributions.

II. RELATED WORK

In the CF technique based on the users, the subset of the users is selected by their similarity, and a weighted combination of their ratings for this user to make production forecast. As an alternative, Linden et al. [3] proposed item-to-item collaborative filtering that rather than matching similar users, then match users rated items to similar items. Both similarity-based approaches do not scale well, computational complexity for the user search because of other similar items. In addition, similar accuracy can be easily by the sparsity.

In order to solve the above mentioned weaknesses community-based traditional CF, Cho et al. [4] proposed a CF model, which hybrid the views of both neighbors and experts, and recommendations made on the basis of the dual information from the both source: a similar-user group and an expert-user group. Further, Amatriain et al. [5] proposed an expert-based collaborative filtering approach, in which the experts defined as an individual who is reasonable to project evaluation. They build the expert database by crawling expert ratings from a web portal called Rotten Tomatoes, and then recommend items to native users based on experts' opinions. Another previous work [6] trying to solve the scalability problem of CF, the first attributed to the user rating data before use traditional stakeholders estimates data most similar neighbors or the densest neighbors CF algorithm.

As a valuable collaborative [7] filtering technology, matrix factorization (MF) model scales well on large scale database. Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, so that user-item

Tao Xu is with the Graduate School of Information, Production and Systems, Waseda University, (e-mail: jyotou@fuji.waseda.jp),
Jing Tian is with the Graduate School of Information, Production and Systems, Waseda University, (e-mail: tasei@ruri.waseda.jp),
Tomohiro Murata is with the Graduate School of Information, Production and Systems, Waseda University, (e-mail: t-murata@waseda.jp).

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

interactions are modeled as inner products in that space. And also the Singular Value Decomposition (SVD) [8] is a well-established technique for identifying latent semantic factors, which is applied in recent works about MF [9], and these models are learned by fitting the previously and the learned model fitting the rating previously observed, while avoiding overfitting learned through the specification parameters.

III. Training Data Set-Based Model

In this article, we use the following symbols. R is denoted in the previously observed rating matrix. R is a $M\times N$ matrix, where M denotes the number of users, and N denotes the number of items. An element r_{ui} of R stores the rating of item i provided by user u, where high values mean stronger preference. The vast majority of ratings in R are missing. U denotes the set of all users, where each user has a row in R indicating his or her ratings of all items. The set of training data sets is denoted by F, which contains H elements, and the training data sets rating matrix R is a $H\times N$ matrix, where $H \ll M$. Rating r_{fi} indicates the preference by film f of item i. In the prediction stage, $\bar{r}_{ui}$ denotes the predicted value of r_{ui}.

A. Main Idea

The Top-N most similar training data sets we called MNNs (most nearly neighbors). In our approach, training data sets are trained based on: (1) users in this set rate more items than the average user; (2) and their ratings would not fluctuate too much. The training process is essentially an information extraction process, through which a large, sparse matrix R is compressed into a small, dense matrix R which contains all the information about latent MNNs or training data sets. Predictions are produced for the active user by combining training data sets' opinions based on their similarity to the active user, which avoid the arduous task of finding the Top-N nearest neighbors of the active user from the large matrix R as we do in traditional neighborhood based methods. Our goal is not to propose a new method that outstrips valuable techniques, but rather to extend the widely used neighborhood-based CF model, so that it can adapt to large-scale data and achieve higher accuracy. In the next subsection we detail the training process of training data sets.

B. Training Data Sets

Our goal is to estimate R, which is a $H\times N$ matrix of parameters. In the training stage, prediction of a given quantities r_{ui} is computed using:

$$\vec{r}_{ui} = \vec{r}_u + \frac{\sum_{f\in F}(r_{fi}-\vec{r}_f)\times w_{u,f}}{\sum_{f\in F} w_{u,f}} \quad (1)$$

where $w_{u,f}$ is the similarity between the user pair (u, f) (where $u \in U$ and $f \in F$), $\vec{r}_u$ is the mean rating given by user u, and similarly $\vec{r}_f$ is the average score given by training data set f. In our model, $w_{u,f}$ is computed using Pearson correlation [11] and we estimate each r_{fi} by minimizing the sum of squared residuals, using gradient descent. The residuals are defined as:

$$e_{ui} = r_{ui} - \left(\vec{r}_u + \frac{\sum_{f\in F}(r_{fi}-\vec{r}_f)\times w_{u,f}}{\sum_{f\in F} w_{u,f}}\right) \quad (2)$$

and the objective function to minimize is given by:

$$Err = \sum_{(u,i)} e_{ui}^2 \quad (3)$$

We applied a simple gradient descent method on the objective function to find a local minimum. The gradient of e_{ui}^2 is:

$$\frac{\partial}{\partial_{r_{fi}}} e_{ui}^2 = -2e_{ui} \cdot \frac{\frac{N-1}{N} \cdot w_{u,f}}{\sum_{f\in F} w_{u,f}} \quad (4)$$

where N is a very large number, so $\frac{N-1}{N}$ is close to 1, which can be ignored in the training process. We update r_{fi} in the direction opposite of the gradient:

$$r_{fi} \leftarrow r_{fi} + \theta \cdot e_{ui} \cdot \frac{w_{u,f}}{\sum_{f\in F} w_{u,f}} \quad (5)$$

where θ is the learning rate.

In our model, the similarity $w_{u,f}$ (for all $u \in U$, $f \in F$) does not need to be learned, we simply update each $w_{u,f}$ using the Pearson correlation every few iterations in the training process. And instead of updating the mean rating $\vec{r}_f$ (for each $f \in F$) whenever matrix R changes, we perform the update of $\vec{r}_f$ in a smoother way to improve training efficiency. See algorithm 1 for the details of training process.

Typically, the number of ratings given by a single user is quite small, thus $\vec{r}_f$, which is the average of the matrix R, changes little after we update elements of matrix R based on a single user's ratings. And thus, we update $\vec{r}_f$ every α passes through the second layer for loop in Algorithm 1.

Algorithm 1 Training Data Sets

Initialize R.
Compute similarity between each ($u \in U$, $f \in F$) pair, and get a relation matrix W.
for iter = 0; iter < maxiter; iter++ do
  for each user $u \in U$ do
    for each rating $r_{ui} \in R$ by user u do
      Compute $\vec{r}_{ui}$
      Compute e_{ui}
      Update r_{fi}
    end for
    Update each mean rating $\vec{r}_f$
  end for
  Update each $w_{u,f} \in W$ every β iterations.
  Validate the model on the test set, if the prediction performance is poor than last iteration, break the outermost for loop.
end for
Return R;

C. Discussion

Amatriain *et* al. [5] external experts predict a small portion of the total population using a recommendation system. However, their approach not increase in the accuracy of the CF. Moreover, they need to select the web to gather experts rated, which will affect the credibility of the sources, mining experts from their final recommendations. By comparison, in our approach, any users who have given rating recommendation system will help training data set training

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

process. Therefore, we believe that the training data set set is more reliable external the crawl experts set, and this can be confirmed by the results of our experiments: as will be seen in section 5, Our approach enjoys a higher accuracy than the traditional neighborhood-based CF, indicating that the training data set eventually provided satisfactory recommendations.

Traditional neighborhood-based CF methods suffer from two limitations: Data sparsity and scalability. On one hand, similar values are based on a common project, so they are unreliable when data are sparse and the common items are few. On the other hand, the computation of similarity costs too much when there is large number of users or items. Our approach has addressed these two shortcomings. Firstly, the training data sets in our model have rated all the items, which ensure that even a lazy user, who has rated few items, can find enough training data set neighbors. Secondly, we reduce the cost of similarity computing from $O(M^2 \cdot N')$ to $O(M \cdot H \cdot N')$, where H is the number of training data sets $(H \ll M)$, and N' is maximum number of ratings given by a single user $(N' \ll N)$. It will decrease the cost of similarity computing, which makes our approach more scalable than traditional ones.

SVD model, as one of the valuable CF techniques, has relatively high recommendation accuracy. However, there are many parameters in the SVD model which need to be manually tuned: (1) two different eventuality distribution functions (pdf_1, pdf_2) for initializing the user feature matrix and the item feature matrix respectively; (2) a real number F to the number of control functions; (3) different learning rates (θ_1, θ_2) and regularization factors (λ_1, λ_2) for users and items. The learning rate and regularization factor are such arbitrary numbers that it is much more difficult to set them well.

Our model has the parameters listed in Table I, it can be relatively easily obtained.

TABLE I
PARAMETERS IN OUR MODEL

Parameter	Explanation
θ	Learning rate
α	Update frequency of $\vec{r}_f$
β	Update frequency of each $w_{u,f} \in W$
H	Number of training data sets
$F(r_{fi})$	pdf for initializing R

IV. RATING BEHAVIOR ANALYSIS

A. Datasets

We have used MovieLens datasets for this experiment. This data set contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service MovieLens and the ratings are made on a 5-star scale, with half-star increments. In our experiments, the whole MovieLens dataset is randomly divided into two parts, of which 80 percent as the training set, and the rest 20 percent as the test set. We randomly divided each dataset with ratio of 8:2 (training data and test data) and experimented on the rating prediction using the item-based approach. In our experiment, the cold-start and non-cold-start condition were mixed, because we used the full ratings of the dataset.

B. Rating Distribution

We analyze the rating behavior of two types of users as training data sets and the MNN users. Our analysis is focus on Average Rating (AVG) and Rating Standard Deviation (STD). In Table II, comparisons of the two user sets are listed in detail.

In pattern recognition, the k-nearest neighbor algorithm (KNN) [10] is a method for classifying objects based on closest k training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification.

The root-mean-square error (RMSE) [11] is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. RMSD is a good measure of accuracy, but only to compare different forecasting errors within a dataset and not between different ones, as it is scale-dependent. These individual differences are also called residuals, and the RMSD serves to aggregate them into a single measure of predictive power.

TABLE II
SUMMARY OF USER SETS

User set	Size	Description
Training Data Sets	200	Trained by our algorithm
MNN Users	200	From KNN

From Figure 1, average rating of the training data sets are centered around 3.7. And over 80 percent of training data sets with an average rating are in the interval between 3 and 4.2. By comparison, MNN users rate movies with an average score smaller than 3.3, and more than 20 percent MNN users rate movies with an average score larger than 4.0. This indicates that, some MNN users may accustom to rate all movies with too high or too low scores while training data sets rate movies with more variability. In the high score area, MNN users' AVG curve almost overlap while in the low score area the percentage of MNN users is slightly larger than all users indicating that even users with relatively strict rating criteria can be selected as MNNs.

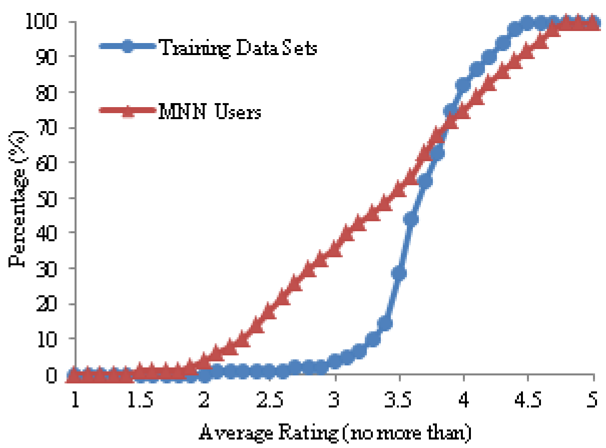

Fig. 1. Distribution of the Average Rating

V. EXPERIMENTAL EVALUATION

We conduct experiments use MovieLens data sets, and the detailed in section 3, to verify the effectiveness of our

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

approach. All of our experiments are executed on a desktop computer, which has 4GB memory and core(TM) I5-2400 CPU of 3.10GHz under the Windows 7 operating system.

A. *Comparing*

TABLE III
PERFORMANCE OF OUR APPROACH COMPARED WITH OTHER MODELS

Model	KNN	SVD	**Our approach**
RMSE	0.9947	0.9203	**0.9025**

We compared the performance of our model with that of other models in Table III. The KNN method produced a RMSE of 0.9947 based on Pearson correlation with k = 20 [12]. Our model with H = 50 makes RMSE reaching 0.9025, which is quite closed the SVD model, which produces one of the best CF techniques with RMSE of 0.9203. Our model and SVD model have trained 100 data sets through training.

B. *Result Analysis*

In our experiments, we initialize each element of training data set rating matrix R, by sampling according to the proportion of 1s to 5s in the previously observed user rating matrix. Here, we choose $\theta = 0.2$, $\alpha = 20$ and $\beta = 5$ for our model in the experiments. For SVD model, we set learning rates for users and items to $\theta = 0.2$, and the regularization factors to $\lambda = 0.05$. All experiments are executed on raw data from MovieLens dataset without any preprocessing.

Figure 2 shows our approach with different H and get the different RMSE values. Our approach with only 20 training data sets got a RMSE = 0.8519. As we can see in Figure 2, the RMSE of our approach is decrease with increasing *H*. However, the larger value of *H*, the slower the convergence of our model. There is no strict standard for setting the value of *H*, we typically find a compromise between accuracy and efficiency. For example, we choose *H* = 170 where RMSE = 0.8359 as the best result of our approach on MovieLens dataset.

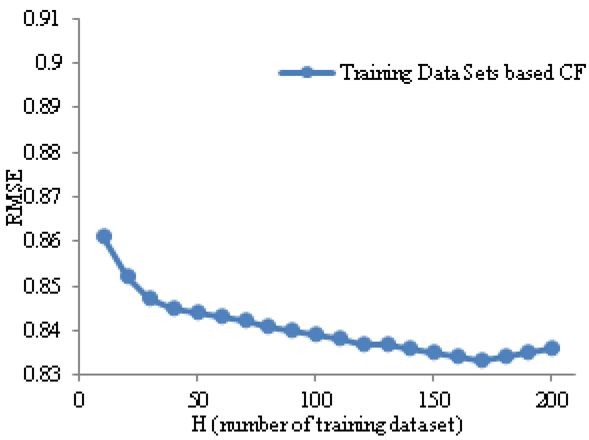

Fig. 2. RMSE of our approach against various H and comparison with KNN

Figure 3 shows our approach in different H and need the different time value. The training our model with H = 20 takes only 179 seconds. Considering both accuracy and efficiency, the value of H can be controlled to a relatively small number, thus even applied on a larger data set our model will not face the problem of unbearably long training time.

By contrast, the KNN model faces a *O(M2 · N')* problem, the increase of user scale M will lead to dramatically increase in the total training time. In addition, the predicting time of KNN model is also much longer than that of our model, as for a specific user, the KNN model needs to find his or her nearest neighbors, which is a very time-consuming process, and time consumed by this process will increase with increasing *k*.

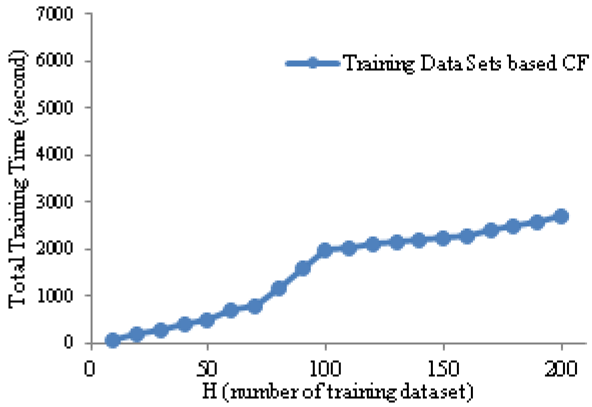

Fig. 3. Time requirement comparison

The best result of SVD model we get on MovieLens dataset is 0.9049, where feature number F is set to 40.

Figure 4 shows the performance of our model with different H values. The parameter H determines the accuracy and efficiency of our model. A too small H value decreases the accuracy of our model, while a too large H value will slow down the convergence process. As is indicated in Figure 4, after training for 100 epochs, our model with H = 50 produced a RMSE of 0.9025.

From all above experiments we conclude that our model provides a relatively efficient and accurate recommendation technique.

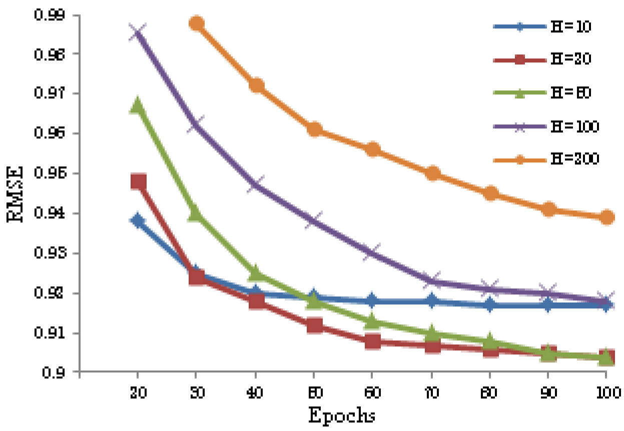

Fig. 4. Our approach with different values of H

From the above mentioned, the value setting of parameter H, which means number of training data set, will affect the value of RMSE.

In Figure 5, it shows that after same epochs, different H will lead almost the same RMSE, especially after 50 epochs. For example, at 100 epochs, with the H value from 10 to 200, RMSE reaches the lowest value at 0.908 and highest value at 0.925, which means that our proposal has better stability.

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

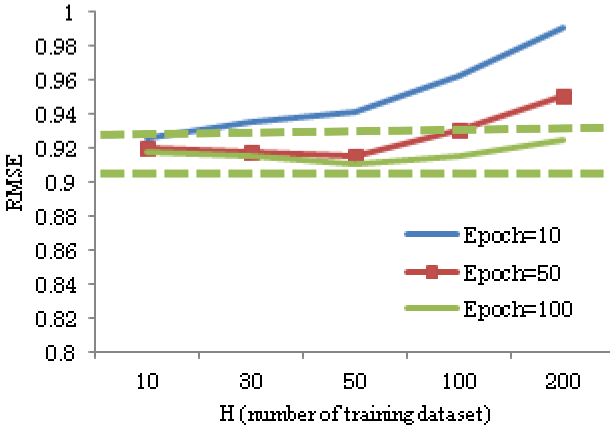

Fig. 5. Our approach with different values of epoch

VI. CONCLUSION

In this paper, we proposed a new collaborative filtering technology, which can make predictions for a large number of users based on only small training data sets. Compared with traditional models, our model is less vulnerable to the data sparsity, meanwhile scales better on large-scale dataset and can solve the cold start problem in better manner. Experiments on a famous dataset named MovieLens datasets demonstrated the effectiveness and efficiency of our proposal.

In future, we attempt to incorporate contextual information of users or items into our model in order to improve the accuracy.

REFERENCES

[1] Middleton, S.E. et al. (2004) Ontological user profiling in recommender systems. ACM Transactions on Information Systems 22(1):54–88.
[2] S. T. Park, W. Chu. Pairwise Preference Regression for Cold-start Recommendation. In ACM RecSys'09, New York, USA, 2009, 21-28.
[3] G. Linden, B. Smith, and J. York, "Amazon com Recommendations: Item-to-item Collaborative Filtering," IEEE Internet computing, vol. 7, no. 1, pp. 76–80, 2003.
[4] H. J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 178 (2008) 37-51.
[5] Y. Li, L. Lu, L. Xuefeng. A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert Systems with Applications, 2005, vol. 28, 67-77.
[6] Billsus, D., and Pazzani, M. J. (1998). Learning Collaborative Information Filters. In Proceedings of ICML '98. pp. 46-53.
[7] SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference (WWW10).
[8] G. H. GOLUB AND C. REINSCH, Singular value decomposition and least squares solutions, Numer. Math., 14 (1970), pp. 403-420.
[9] R. M. Bell, Y. Koren and C. Volinsky, "Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems", Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.
[10] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In Proc. CVPR, 2006.
[11] P.E. and ROBERTS, D.A., 2003, Endmember selection for multiple endmember spectral mixture analysis using endmember average RMSE. Remote Sensing of Environment, 87, pp. 123–135.
[12] R. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," in KDD-Cup and Workshop. Citeseer, 2007.