

Recommendation using DMF-based fine tuning method

Zhiyuan Zhang^{1,2}, Yun Liu^{1,2,*}, Guandong Xu³, Guixun Luo^{1,2}

1, School of Communication and Information Engineering, Beijing Jiaotong University, Beijing 100044, China.

2, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China.

3, Advanced Analytics Institute, University of Technology Sydney, Australia.

* Corresponding author. e-mail: liuyun@bjtu.edu.cn.

Abstract

Recommender Systems (RS) have been comprehensively analyzed in the past decade, Matrix Factorization (MF)-based Collaborative Filtering (CF) method has been proved to be an useful model to improve the performance of recommendation. Factors that inferred from item rating patterns shows the vectors which are useful for MF to characterize both items and users. A recommendation can concluded from good correspondence between item and user factors. A basic MF model starts with an object function, which is consisted of the squared error between original training matrix and predicted matrix as well as the regularization term (regularization parameters). To learn the predicted matrix, recommender systems minimize the squared error which has been regularized. However, two important details have been ignored: (1) the predicted matrix will be more and more accuracy as the iterations carried out, then a fix value of regularization parameters may not be the most suitable choice. (2) the final distribution trend of ratings of predicted matrix is not similar with the original training matrix. Therefore, we propose a Dynamic-MF(DMF) and DMF-based fine tuning method which is quite general to overcome the mentioned detail problems. Some other information, such as social relations, etc, can be easily incorporated into this method (model). The experimental analysis on two large datasets demonstrates that our method outperform the basic MF-based method .

Key Words: Recommender Systems, Matrix Factorization, Collaborative Filtering, Fine Tuning, Dynamic, Social Relations

1 Introduction

Recommender Systems (RS) are computer applications and techniques for recommending specific items that may meet users' preference. Recommender systems can be generally classified into content-based (CB) (Bobadilla et al. 2013; Melville et al. 2002) Recommendation , collaborative filtering (CF) recommendation (Resnik et al. 1994; Sarwar et al. 2001; Massa and Avesani. 2009), and hybrid recommendation (Adomavicius et al. 2005; Lu et al. 2012). CB recommendation tries to recommend items that are similar to the ones which are of users' historical preference. This recommend method usually extract useful

external information, such as user profiles (Zhang et al. 2014), features, explicit item descriptions (Marin et al. 2013), etc, to describe user preference to provide recommendation. CF recommendation is a practical method adopted by mainstream RS. The core idea of CF is that similar users express similar preference, which means an item can be recommended to a target user by analyzing his similar users' preference. The most important point of CF is that it usually relies on the past user-item rating history information to describe and create a model. The hybrid recommendation combines both CB-based and CF-based method to improve the performance of RS and avoid certain limitations of single RS.

CF is the most widely used technique for Recommender Systems (Massa et al. 2004). It is so important as a heat pot that lots of research work are attracted to contribute to the task. There are two primary areas of collaborative filtering named the neighborhood methods (Zhang et al. 2006; Forsati et al. 2014) and latent factor models (Koren et al. 2009). Neighborhood methods measure the similarity (similarity of users, similarity of items, similarity of combination) based on ratings of items given by users. Neighborhood methods are very effective at finding local similarity. Detecting neighborhood relationships(similarity) is the key point of those methods. However, neighborhood-based CF suffers some weaknesses: cold start issues and data sparsity (Lu et al. 2012). When new users enter the system, there is usually insufficient information to produce recommendation for them, that is so called cold start problem. Data sparsity implies that there is not enough data to analyze similarity between target users. To overcome the limitations, latent factor models are proposed (Forsati et al. 2014). Latent factor models comprise an alternative approach by transforming both items and users to the same latent factor space, thus making them directly comparable. Latent factor models are generally effective at estimating overall structure that relates simultaneously to most or all items (Koren. 2008). Establishing an appropriate model using the observed ratings/ evaluations is the key point to interpret the given data and to predict the unknown ratings/evaluations. Among all the latent factor models, matrix factorization-based methods have recently received greater exposure (Bokde et al. 2015) and have been focused more widely and proved to be an effective method.

During various kinds of input data, explicit feedback are the most convenient data for RS. Ratings/Evaluations are the most obvious explicit information, which can reflect users' interest in products. These ratings are placed into a big matrix with one dimension representing users and one dimension representing items/products. Generally, the matrix is of high levels of sparsity i.e., many values in rating matrix are null since users are likely to have rated a fraction of items. MF algorithm decompose the user-item matrix into a user potential factor matrix and an item potential factor matrix by using dimension reduction technology (Bobadilla et al. 2013; Bokde et al. 2015), then the predicted matrix can be computed by dotting the two factor matrix.

This paper mainly study the basic MF models, which map both users and

items to a joint latent factor space of a low dimensionality. That means the original matrix will be factorized to two low-dimension latent matrices, which are used to compute predicted matrix. Some very effective matrix factorization based methods, such as Singular Value Decomposition (SVD) model (Liu et al. 2015), Matrix Factorization (MF) (Koren et al. 2009), Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih. 2008b), Non-negative Matrix Factorization (NMF) (Lee et al. 1999, 2001), use an objective function corresponding to the root square error of final predicted matrix with respect to original training matrix to indicate the difference between the two mentioned matrixes . The predicted matrix can be found by minimizing the objective function (Please see section 2 for more details). The SVD model is a powerful technical of dimensionality reduction. How to find a proper lower dimensional feature space is the key issue of SVD model. PMF is a probabilistic linear model with Gaussian observation noise, it models the rating matrix as a product of two low-rank matrices (users and items). Probabilistic Sparse Matrix Factorization (PSMF) (Dueck et al. 2004), Bayesian Probabilistic Matrix Factorization (BPMF) (Liu et al. 2013; Salakhutdinov and Mnih. 2008a), General Probabilistic Matrix Factorization (GPMF) (Shan and Banerjee, 2010) are all the effective probabilistic models. Non-negative Matrix Factorization is also called non-negative matrix approximation, which is greatly developed by Lee and Seung (1999). In NMF model, the original rating matrix is factorized into two matrices, with the property that all values of the three matrices are no-negative.

There are two parts of object function of MF model, root square error between original matrix and predicted matrix and regularization terms. Regularization terms constraint the latent matrices for users and items to prevent over-fitting. In the latest phase of the research, Adding useful related information to regularization terms has become a hotspot of the field of MF. For example, (Ma et al. 2011a) propose a method of MF combining social regularization. (Zhang and Liu, 2015) present a social recommendation model combining trust propagation and sequential behaviors. (Forsati et al. 2014) propose a MF-based model that properly incorporates both explicit trust and distrust side information in order to improve the performance of social recommendation.

In basic MF model, both regularization parameters is a fixed value. However the predicted factor matrix will be more and more accuracy based on iterations, then a fix value of regularization parameters may not be the most suitable choice (Please see section 3 for more details). Based on the above analysis, a Dynamic-MF is proposed in this paper. Furthermore, a method of fine tuning is put forward by observing and analyzing the distribution of ratings of original training matrix and ratings of the final predicted matrix (shown in section 4).

The rest of this article is organized as follows: Section 2 introduces the basic MF model. Section 3 states the proposed DMF method. The method of fine tuning is presented in Section 4, followed by experiments and results in Section 5. Finally, Section 6 concludes this paper and put forward the vision of future work.

2 Basic MF model

Usually, the behaviors of users can be modeled in a big user-item matrix (Luo et al. 2014; Koren et al. 2009; Ma et al. 2011a; Ma et al. 2011b). Like we said before, the matrix was very sparse, and the modeled explicit ratings express users' preference about items. Then the missed data (ratings) of this user-item rating-matrix can be predicted by MF-based CF method. In this section, we introduce the traditional MF algorithm.

We assume that $U = \{u_1, u_2, \dots, u_n\}$ stands for a set of n users and $I = \{i_1, i_2, \dots, i_m\}$ stands for a set of m items. Then the whole user-item matrix can be defined as $R^{n \times m}$, where the rows correspond to users and columns correspond to items, r_{ij} is the numerical preference (rating) of user i on item j , $1 \leq i \leq n$, $1 \leq j \leq m$. The method of MF to recommender system is to factorize the user-item matrix R to two low-dimensional matrices $P^{n \times f}$ and $Q^{f \times m}$, Where $P^{n \times f}$ and $Q^{f \times m}$ stand for user-specific and item-specific matrices, respectively, and f is f -dimensional specific potential feature of user and item, usually, $f \ll n, m$. The MF formula is $R^{n \times m} \approx P^{n \times f} Q^{f \times m}$. User-item matrix R is very spare in most cases, then the method of MF make it possible to effectively predict and recover the missing data of the rating matrix by learning the observed ratings.

From the above definitions, our purpose is to learn the most accurate f -dimensional matrix P and Q to estimate the missing entries of the big spare user-item matrix R . We define the estimated matrix as $\hat{R}^{n \times m} = P^{n \times f} Q^{f \times m}$, so $\hat{R}^{n \times m} \approx R^{n \times m}$, then the missing entries of R can be filled by \hat{R} in every training iteration. This factorization problem can be solved by minimizing the difference between the original matrix R and the estimated matrix \hat{R} . This difference is defined by Euclidean distance, so the MF-based CF problem is given as follows:

$$\min \left[d = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m I_{ij}^R (R_{ij} - \sum_{k=1}^f P_{ik} Q_{kj})^2 + \frac{\lambda_P}{2} \|P\|_F + \frac{\lambda_Q}{2} \|Q\|_F \right], \quad (1)$$

Eq.(1) is the object function, where $\|\bullet\|$ is the Frobenius norm of a matrix, that is,

$$\|P\|_F = \sqrt{\sum_{a=1}^n \sum_{b=1}^f |P_{ab}|^2} \cdot \frac{\lambda_P}{2} \|P\|_F \quad \text{and} \quad \frac{\lambda_Q}{2} \|Q\|_F \quad \text{are regularization terms. } I_{ij}^R \text{ is the}$$

indicator function that is equal to 1 if user i rates item j and equal to 0 otherwise. The parameters λ_p and λ_Q are regularizing coefficients for P and Q respectively, which are used to prevent over-fitting. This is a non-convexity optimization problem, and it can be solved by conventional optimization algorithms (Koren et al. 2009; Ma et al. 2011b). For instance, stochastic gradient descent is an effective method to optimize Eq.(1), each desired element is obtained via the following training process:

$$\begin{cases} P_{ik} \xleftarrow{\text{update}} P_{ik} + \partial [I_{ij}^R (RQ^T)_{ik} - I_{ij}^R (PQQ^T)_{ik} - \lambda_p \cdot P_{ik}] \\ Q_{kj} \xleftarrow{\text{update}} Q_{kj} + \partial [I_{ij}^R (P^T R)_{kj} - I_{ij}^R (P^T PQ)_{kj} - \lambda_Q \cdot Q_{kj}] \end{cases}, \quad (2)$$

where ∂ is the corresponding learning-rate. P_{ik} denotes the corresponding entry(rating) of P at row i and column k , Q_{kj} denotes the corresponding entry(rating) of Q at row k and column j , as mentioned before, these two parameters actually denote the k th latent feature of user u_i and the k th latent feature of item i_j , respectively. Initially, entries of P and Q are random, during the training process they may be updated to more accurate values, in other words, $R \approx \hat{R}$. The whole process is shown in Figure 1. The above object function is a baseline function of MF-based CF method, some other regularization terms can be added to constraint the object function, such as social factors (Kim and Chen. 2015; Zhang and Liu. 2015; Qian et al. 2014; Sherchan et al. 2013; Wierzowiecki et al. 2010). In this paper, we focus on the original object function, especially the parameters. We hope to make it more efficient to system architectures for recommender systems.

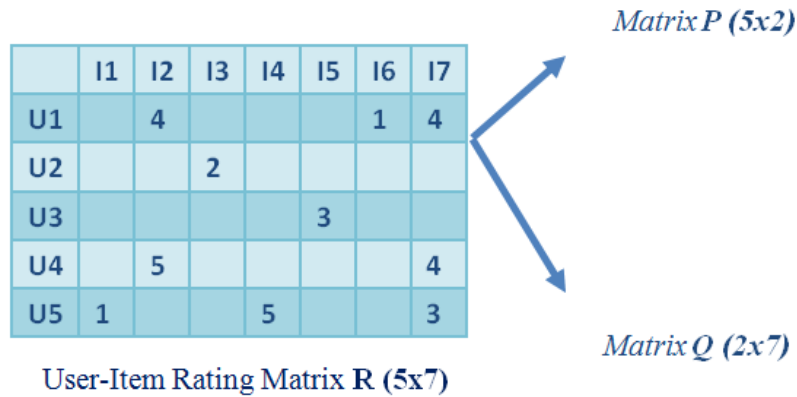


Fig. 1 Example of the process of MF

3 Dynamic-MF for CF problems

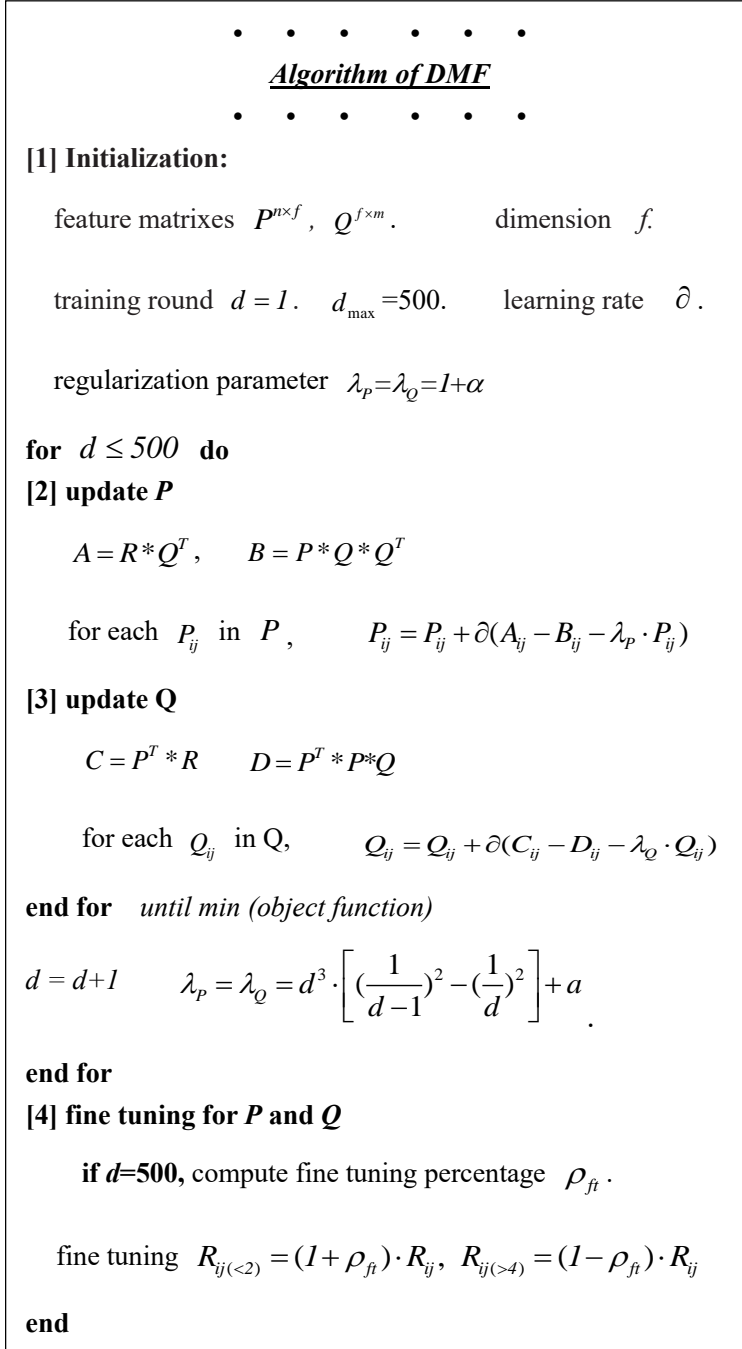


Fig. 2 DMF-based fine tuning method

As discussed in section 2, traditional MF algorithm to recommender system consider the rating-matrix estimate in a CF problem. In Eq.(1) consists of three parts (Forsati et al. 2014). The first part is the Euclidean distance between user-item matrix R and the f -dimensional estimate PQ , which should be minimized through the method of optimization in every iteration. The last two parts constraint the latent matrices for users and items, respectively as

regularization terms. The parameters λ_p and λ_q are regularization parameters that are introduced to control the regularization of the trained latent matrices P and Q .

In traditional MF-based CF, both regularization parameters is the same fixed perfect value. But in the sense of experience, the estimated $PQ (\hat{R})$ will be more and more accurate, the previous papers (Koren et al. 2009; Luo et al. 2014; Ma et al. 2011b) prove that the final results are convergent and \hat{R} is more and more accurate to make E.q. (1) minimized. Obviously, \hat{R} is not fix in every iteration, so it reminds us that the regularization parameter should not be a fix value to control the regularization of P and Q in every iteration, because the accuracy of different P and Q make different levels of contribution to the object function and the result of next iteration. Every 20 iterations the predicted matrix is collected as a standard of accuracy, thus we totally collect 11 original training matrix (0, 20, 40, ..., 200) with different accuracy. In order to find the change rule of regularization parameters, every matrix will be tested using 5 regularization parameter values (5, 4, 3, 2, 1), then one of these values will be the best regularization parameter value for one of 11 matrices. Figure 3 shows us the change rule (yellow line) of regularization parameters in the case of matrix of different accuracy. We can draw a basic conclusion that the more accuracy of \hat{R} , the less value of regularization parameters .

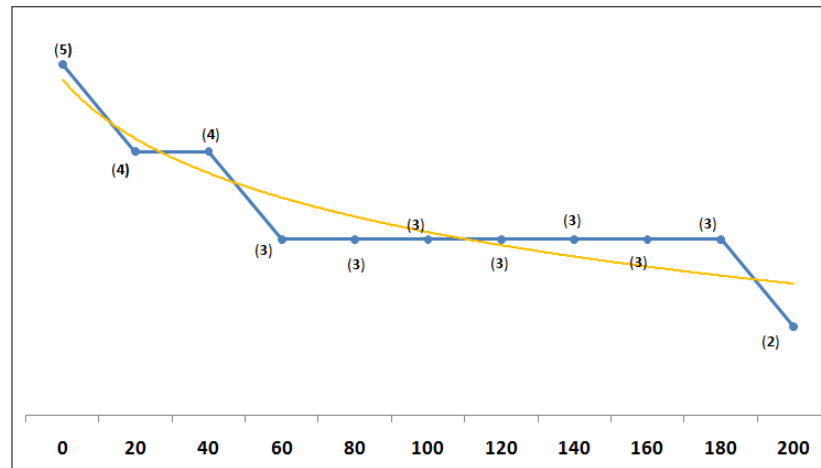


Fig. 3 regularization parameters changing with the matrix of different accuracy (change rule of regularization parameters)

In our method, regularization parameters are not a fix value any more, it is a dynamic parameter that can change dynamically. We define the dynamic parameter as a function of the number of iterations, given by:

$$\lambda_p = \lambda_Q = \begin{cases} d^3 \cdot \left[\left(\frac{1}{d-1} \right)^2 - \left(\frac{1}{d} \right)^2 \right] + a & \text{s.t. } d > 1 \\ 1 & \text{s.t. } d = 1 \end{cases}, \quad (3)$$

where d denotes the number of iterations. When the first time of iteration, λ_p and λ_Q can be initialized to $\lambda_p = \lambda_Q = d^2 \cdot \frac{1}{d^2} + a$. a is the constrict factor, which has two functions, the first one is to control the parameter to a base line. So the value of parameter will not too small in every iteration. The second one is in order to fit different kinds and densities of datasets. Finally, the dynamic regularization parameters can be used in E.q. (2). We call this method as Dynamic-MF, or DMF. The effect of DMF and how to select a better a will be shown in section 5.

4 Process of fine tuning

Like we said before, matrix $R \approx \hat{R}$, the missing entries are predicted by learning the matrix R , so the matrix \hat{R} is an updated full matrix, which can provide more useful information to complete the process of recommendation. When the final result of \hat{R} obtained by MF-based method, we can recommend new items, which were not rated or didn't pay attention by users, to users according to the predicted ratings in \hat{R} .

However, we should not only pay close attention to the predicted ratings, but also put ourselves into analyzing the character of the whole updated matrix \hat{R} . For example, if ratings range from 1 to 5, Figure 4 describes the distribution of original ratings of test datasets and the final predicted ratings of datasets (datasets come from Movielens and Epinions, which will be described detailedly in the next section). We can conclude an interesting observation that the predicted ratings focus on the range of 2-4, but the number of 1 and 5 are very little. Obviously, it is inconsistent with the trend of the real situation. That means some rating score 1 or 5 were updated into range 2-4. So it is necessary to make a fine tuning after getting the predicted matrix \hat{R} . The ratings less than 2 should be decreased meanwhile the ratings more than 4 should be increased in a proper range to make the distribution of predicted matrix similar to the original training matrix.

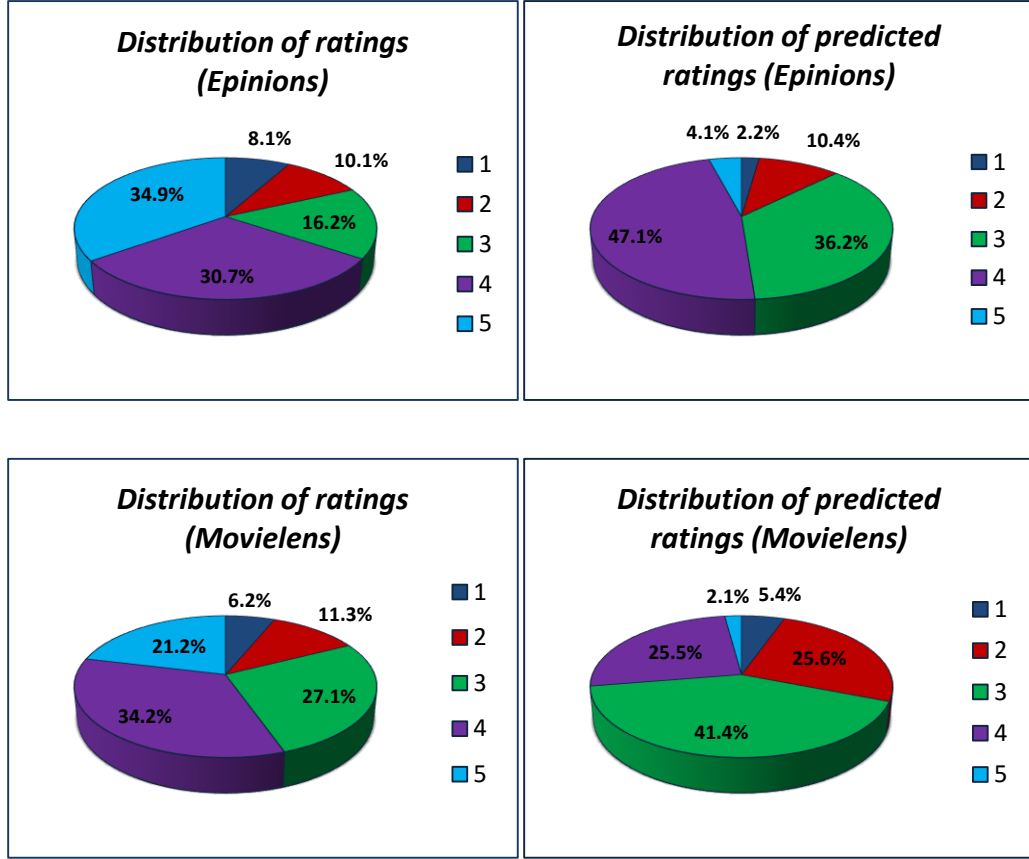


Fig. 4 Distribution of original ratings (left) and predicted ratings (right) .

Only we have now is the original training matrix and the final predicted matrix. The fine tuning percentage is given by:

$$\rho_{ft} = \left[100 \cdot \frac{\left| \left(\sum_{i,j \in \Theta_R} R_{ij} - \sum_{i,j \in \Theta_{\hat{R}}} \hat{R}_{ij} \right) \right|}{|\Gamma|} \right] \% , \quad (4)$$

where Θ_R is the set of observed ratings of R , that is:

$$\Theta_R = [(i, j) \in [n] \times [m] : R_{ij} \neq \text{empty}].$$

$\Theta_{\hat{R}}$ is the set of predicted ratings corresponding to the same position with R ,

which means R_{ij} and \hat{R}_{ij} have the same coordinate " ij ". $|\Gamma|$ denotes the number of ratings of original training matrix R .

Then the ratings less than 2 and more than 4 should be updated as:

$$\left\{ \begin{array}{l} \hat{R}_{ij \leftarrow \text{update}} = (1 + \rho_{ft}) \cdot \hat{R}_{ij} \quad \text{s.t. } \hat{R}_{ij} < 2 \\ \hat{R}_{ij \leftarrow \text{update}} = (1 - \rho_{ft}) \cdot \hat{R}_{ij} \quad \text{s.t. } \hat{R}_{ij} > 4 \end{array} \right. , \quad (5)$$

Figure 2 illustrates pseudo-code of the algorithm of DMF and the process of fine tuning.

5 Experiments and Results

5.1 Datasets

To test the methods, we use two standard public datasets: Movielens 1M (Zhang et al. 2014) and Epinions[Ma et al. 2009a, 2009b].

Movielens datasets contains ratings by users on their watched movies to form a large user-movie rating matrix, the rating score range from 1 to 5. For our experiments, we use a 100000 ratings matrix to check whether the proposed method can bring high efficiency and accuracy, the rating density of selected datasets is 6.3%. We call the sampled datasets D1.

Epinions.com¹ is a well known knowledge sharing site and review site, which was established in 1999. User can assign movies, reviews or some others products, which was selected from social network, ratings from 1 to 5. So the users and items and ratings are able to form a large sparse user-item rating matrix. The dataset used in our experiments consists of 1000000 ratings and the density of the selected user-item matrix is 1.5%. We can observe that the user-item matrix of Epinions is very sparse. The fact that other part of the dataset also contains explicit positive and negative relations between users makes it very appropriate for future studying issues in social-based recommender systems, we just care about the rating matrix in this paper. We call this sampled datasets D2.

Datasets	User	Item
Epinions	943	1628
Movielens	5538	12038

Table 1. Statistics of User-Item Rating Matrix

The statistics of data source is summarized in Table 1. Obviously, D1 and D2 are different kinds of experiment datasets. They have different densities and they collect different opinions (ratings) by different users on different kinds of items. Although the rating matrices are formed by integer numbers, they come from different social network or e-commerce network.

5.2 Metrics

Two well-known measures, mean absolute error (MAE) and root mean squared error (RMSE) (Lu et al. 2012), were employed to compute the prediction accuracy of proposed method.

In order to measure the accuracy of the results of an RS, we usually use the calculation of some of the most common prediction error metrics, among which the Mean Absolute Error (MAE) and its related metrics: mean squared error, root mean squared error, and normalized mean absolute error stand out. MAE is calculated by measuring the difference between the predicted ratings and the real ratings (test datasets), the errors should be averaged over all predicted ratings to

¹ WWW.Epinions.com

obtain a mean final value. MAE is defined as:

$$MAE = \frac{\sum_{i,j \in \Omega_\eta} |R_{ij} - \hat{R}_{ij}|}{|\eta|},$$

More precisely, $|\eta|$ denotes the number of ratings of test datasets R_{test} , " ij " denote the coordinate of rating in R_{test} , that is:

$$\eta = \{(i, j) \in [n] \times [m], R_{ij} \in R_{test}, \hat{R}_{ij} \in \hat{R}\},$$

where \hat{r}_{ij} need to be predicted by the algorithms.

RMSE measures larger errors compare to MAE (mean error), RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j \in \Omega_\eta} (R_{ij} - \hat{R}_{ij})^2}{|\eta|}},$$

where the elements $(R_{ij}, \hat{R}_{ij}, \eta)$ of RMSE have the same meaning of MAE. From the formulas, we can draw a conclusion that lower MAE and RMSE value means higher prediction accuracy.

All tested models are implemented in MATLAB R2012a, and tested on a PC Server with a 2.2 GHz CPU and 8 GB Memory.

5.3 Experimental Process

In order to test the effectiveness of the proposed method, the traditional MF model described in section 2 and some other state-of-arts methods are used as the baseline methods. A cross-validation technique will be used in the paper. We randomly select 90%, 80%, 70%, 60%, and 50% of D1 and D2 as the training-sets to validate the performance. The random selection was carried out 5 times independently.

As descriptions in section 3, in DMF, the constrict factor α control the extent of regularization under the condition of dynamic change of regularization parameters. Prediction accuracy and convergence rate will be affected by this factor. Therefore, different values are verified to find a reasonable value for the social regularization constrict factor.

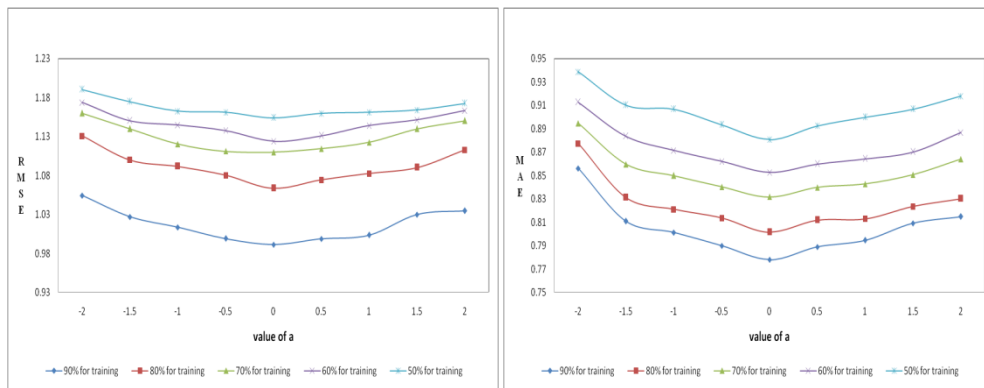
On both two datasets, we validated the performance of DMF (section 3) and DMF-based fine tuning method (section 4) to compare with traditional MF and NMF method. For a fair comparison, the training process of each model contains 500 training iterations (convergence state) on each dataset.

5.4 Results and Discussions

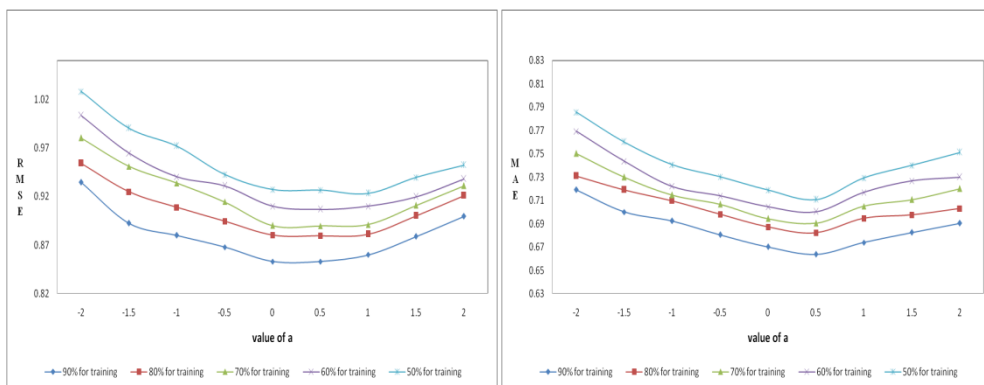
In this section , we discuss two experimental results. Firstly, the impact of factor α is shown in part 5.4.1. Secondly, the comparison of some state-of-arts methods will be shown in part 5.4.2, we can easily find the performance of the method of DMF-based fine tuning from the comparisons.

5.4.1. Impact of factor α

E.q. (2), (3) are used in the process of the experiment. We use different values of α (-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2) to complete the experiment on both datasets. When it achieves the final predicted matrix \hat{R} , the best RMSE and MAE value on each datasets of different α are shown in Figure 5. We see that α has vital effect on prediction accuracy of DMF. We can find with $\alpha=0$, the algorithm gets the best performance of prediction accuracy on datasets Epinions. With $\alpha=0.5$, the algorithm gets the best performance of prediction accuracy on datasets Movielens. The phenomenon demonstrates when the constrict factor α in the range from -0.5 to 0.5, a better RMSE and MAE will be obtained on datasets Epinions, For the datasets Movielens of this experiments, the better rang of constrict factor α is from 0 to 0.5.



(1) Epinions



(2) Movielens

Fig. 5 Impact of different value of constrict factor α on performance.

5.4.2. Performance of DMF based fine tuning method

Four methods are tested in the experiment. First, the traditional MF, which is

described in section 2, is implemented and tested as a baseline method. Secondly, the NMF (Lee et al. 1999, 2001) is also tested as another benchmark. Note that we choose NMF as benchmark is because it is another widely used method to solve CF problem. Moreover, It provides us another idea to learn matrices P and Q , each value of matrices P and Q is nonnegative. Then the proposed DMF and DMF-based fine tuning methods are compared with MF and NMF to validate if our strategies can bring positive effect on performance.

As discussed, for the datasets Epinions, $\alpha=0$ will be used in DMF and DMF-based fine tuning method to get the best RMSE and MAE. For the datasets Movielens, $\alpha=0.5$.

	Method	90% for training	80% for training	70% for training	60% for training	50% for training
RMSE	MF	0.8896	0.8973	0.9086	0.9238	0.9491
	NMF	0.8788	0.8898	0.9015	0.9197	0.9418
	DMF	0.8645	0.8801	0.8993	0.9125	0.9386
	DMF ft	0.8529	0.8796	0.8897	0.9066	0.9265
MAE	MF	0.6997	0.7072	0.7189	0.7298	0.7376
	NMF	0.6843	0.6908	0.7012	0.7198	0.7288
	DMF	0.6787	0.6802	0.6927	0.7069	0.7148
	DMF ft	0.6696	0.6762	0.6834	0.6932	0.7086
<i>(1) Movielens</i>						
	Method	90% for training	80% for training	70% for training	60% for training	50% for training
RMSE	MF	1.0361	1.0954	1.1203	1.1505	1.1732
	NMF	1.0269	1.0868	1.1194	1.1462	1.1649
	DMF	1.0029	1.0711	1.1143	1.1308	1.1608
	DMF ft	0.9943	1.0636	1.1098	1.1240	1.1540
MAE	MF	0.8006	0.8398	0.8579	0.8926	0.9147
	NMF	0.7865	0.8245	0.8482	0.8803	0.9025
	DMF	0.7742	0.8123	0.8413	0.8611	0.8897
	DMF ft	0.7613	0.8084	0.8337	0.8527	0.8804
<i>(2) Epinions</i>						

Table 2, RMSE and MAE comparison of MF, NMF, DMF and DMF-based fine tuning method

From Table 2, we can observe that our method outperforms the other models. In general, the proposed DMF and DMF-based fine tuning method both perform better than MF and NMF on both RMSE and MAE.

6 Conclusion and future work

In this article, we focus on improving the recommend efficiency of MF-based CF

method. Firstly, we propose a Dynamic-MF algorithm based on the traditional MF model. In traditional MF model, they use a fixed value as the regularization parameters to learn and update target matrix, but they neglect an important phenomenon that the predicted matrix is more and more accurate based on iterations. So regularization parameter should not be a fixed value but an updated value based on iterations. Secondly, by analyzing the distribution trend of ratings of original training matrix and predicted matrix, we find that the proper distribution trend of ratings of predicted matrix should be similar with the training matrix. Hence, we introduced a fine tuning method according to the difference between predicted matrix and original training matrix. Based on the experimental analysis, The proposed DMF-based method leads to improved prediction accuracy, after getting the predicted matrix by DMF, the fine tuning method proved to have a better trend of results.

MF-based method has become popular in recent years by combining good scalability with predictive accuracy. In this paper, we only constrain the adjustment of parameters on the level of system architecture while ignoring some important regularization terms, such as users' social relation information, correlations between items, etc. In future work, the mentioned information can be incorporated into our framework to improve the performance of recommendation. Furthermore, other metrics can be used to test the effectiveness of the specific recommend system.

Acknowledgement

This work has been supported by the Fundamental Research Funds for the Central Universities 2016YJS028 and the National Natural Science Foundation of China under Grant 61172072, 61401015, 61271308.

References

- Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17 (6) 734–749.
- Bobadilla, J., Ortega, F., Hernando, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109-132.
- Bokde, D., Girase, S., Mukhopadhyay, D. (2015). Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. In *Proceedings of 4th International Conference on Advances in Computing, Communication and Control* (pp.136-146).
- Dueck, D., Frey, B, J. (2004). Probabilistic Sparse Matrix Factorization. *Technical Report PSI TR*, 2004-023.
- Forsati, R., Mahdavi, M., Shamsfard, M., Sarwat, M. (2014). Matrix Factorization with Explicit Trust and Distrust Side Information for Improved Social Recommendation. *ACM Transactions on Information Systems*, 32(4), 1-38.
- Kim, M, C., Chen, C. (2015). A scientometric review of emerging trends and new developments in recommendation systems. *Scientometrics*, 104, 239-263.

- Koren, Y. (2008). Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp.426-434)
- Koren, Y., Bell, R., Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), 30-37.
- Lee, D. D., Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788-791.
- Lee, D. D., Seung, H. S. (2001). Algorithms for Non-negative Matrix Factorization. In *Proceedings of Neural Information Processing Systems*. 32(6):556--562
- Liu, J. T., Wu, C. H., Liu, W. Y. (2013). Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation. *Decision Support Systems*, 55(3), 838–850.
- Liu, W., Wu, C. H., Feng, B., Liu, J. T. (2015). Conditional preference in recommender systems. *Expert Systems with Applications*. 42(2), 774-788.
- Lu, L. Y., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K. (2012). Recommender systems. *Physics Reports*, 519(1), 1-49.
- Luo, X., Zhou, M., Xia, Y., Zhu, Q. S. (2014). An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Transactions on Industrial Informatics*, 10(2), 1273-1284.
- Ma, H., King, I., Lyu, M. R. (2009a). Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp.203-210).
- Ma, H., Lyu, M. R., King, I. (2009b). Learning to Recommend with Trust and Distrust Relationships. In *Proceedings of the 3rd ACM Conference on Recommender Systems* (pp.189-196).
- Ma, H., Zhou, D. C., Lyu, M. R., King, I. (2011b) Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst*, 29(2), 219-230.
- Ma, H., Zhou, D., Liu, C., Lyu, M. R., King, I. (2011a) Recommender Systems with Social Regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining* (pp.287-296).
- Marin, L., Isern, D., Moreno, A. (2013). Dynamic adaptation of numerical attributes in a user profile. *Applied Intelligence*, 39(2), 421-437.
- Massa, P., Avesani, P. (2004). Trust-aware Collaborative Filtering for Recommender Systems. *Lecture Notes in Computer Science*, 492-508.
- Massa, P., Avesani, P. (2009). Trust metrics in recommender systems. In *Computing with Social Trust* (pp.259-285).
- Melville, P., Mooney, R. J., Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence* (pp.187-192).
- Qian, X., Zhao, G., Mei, T. (2014). Personalized Recommendation Combining User Interest and Social Circle. *IEEE Transactions on Knowledge and Data Engineering*, 26(7), 1763-1777.
- Resnik, P., Iacovou, N., Suchak, M., Bergsorm, P., Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM*

- Conference on Computer Supported Cooperative Work* (pp.175–186).
- Salakhutdinov, R., Mnih, A. (2008a). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning* (pp.880-887).
- Salakhutdinov, R., Mnih, A. (2008b). Probabilistic Matrix Factorization. In *Proceedings of the 22nd Annual Conference on Neural Information Processing System* (pp.1257-1264).
- Sarwar, B., Karypis, G., Konstan, J., Reidl, J. (2001). Item based collaborative filtering recommendation algorithms. In *Proc. 10th Int. Conf. World Wide Web*, (pp. 285–295).
- Shan, H., Banerjee, A. (2010). Generalized Probabilistic Matrix Factorizations for Collaborative Filtering. In *The 10th IEEE International Conference on Data Mining* (pp.1025-1030).
- Sherchan, W., Nepal, S., Paris, C. (2013). A survey of trust in social networks. *ACM Computing Surveys*, 45(4), 115-123.
- Wierzowiecki, G., Wierzbicki, A. (2010). Efficient and correct trust propagation using closelook. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (pp.676-681).
- Zhang, J., Peng, Q., Sun, S., Liu, C. (2013). Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Physica A Statistical Mechanics & Its Applications*, 396(2), 66–76.
- Zhang, S., Wang, W, H., Ford, J., Makedon, F. (2006). Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 6th slam Conference on Data Mining* (pp. 549-553).
- Zhang, Z, Y., Liu Y., Qing A Z. (2014). A weighted method to update network user preference profile dynamically. *International Journal of Interdisciplinary Telecommunications and Networking*, 6(2), 52-67.
- Zhang, Z., Liu, H. (2015). Social recommendation model combining trust propagation and sequential behaviors. *Applied Intelligence*, 43(3), 695-706.