A movie recommendation model combining time information and probability matrix factorisation

Huali Pan and Jingbo Wang

Business School,
Shandong Normal University,
Jinan, SD 250014, China
Email: 909461659@qq.com
Email: 928957616@qq.com

Zhijun Zhang*

School of Computer Science and Technology,
Shandong Jianzhu University,
Jinan, SD 250101, China
Email: zzjsdcn@163.com
*Corresponding author

Abstract: A deep analysis and discussion of matrix factorisation technologies are given in this paper taking into account the defects of traditional collaborative filtering recommendation algorithms. In addition, we provide an analysis of the effects of feature vector dimensions on the recommendation quality and efficiency of a probability matrix factorisation (PMF) algorithm. A PMF algorithm will lead to inaccurate recommendations if it does not consider possible dynamic changes in a user's interest over time. Accordingly, a TPMF model, a PMF algorithm integrated with time information, is proposed in this article. Its feasibility and effectiveness are empirically verified using movie recommendation datasets, and higher prediction accuracy is confirmed compared to existing recommendation algorithms.

Keywords: collaborative filtering; matrix factorisation; movie dataset; personalised recommendation; time information.

Reference to this paper should be made as follows: Pan, H., Wang, J. and Zhang, Z. (2021) 'A movie recommendation model combining time information and probability matrix factorisation', *Int. J. Embedded Systems*, Vol. 14, No. 3, pp.239–247.

Biographical notes: Huali Pan received her PhD degree in Economics from Shandong Normal University, Jinan, in 2013. Since 2003, she has been a teacher in Shandong Normal University. And since 2011, she has been an Associate Professor in the Department of Tourism Management, Business School at Shandong Normal University, Jinan, China. She visited Oklahoma State University as a Visiting Scholar from 2015 to 2016. She is the author of three books, more than 20 articles, and more than 30 projects. His research interests include hospitality and tourism management, tourism economics and tourism planning.

Jingbo Wang received her Bachelor's degree in Management from the Inner Mongolia University of Finance and Economics in 2017. She is currently pursuing a Master's degree in Business School of Shandong Normal University. Her research direction is tourism management. She usually does some projects follow her instructor and writes some articles about management.

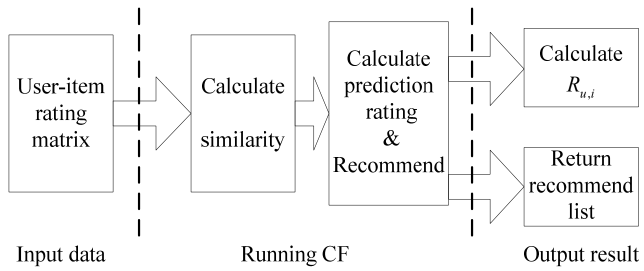
Zhijun Zhang received his PhD degree in School of Information Science and Engineering, Shandong Normal University in 2015. From 1997 to 2020, he was a teacher in Shandong Jianzhu University. Since 2017, he has been a Professor in the School of Computer Science and Technology at Shandong Jianzhu University, Jinan, China. He visited Oklahoma State University as a Visiting Scholar from 2015 to 2016. Currently, his recent research interests include information retrieval and recommender systems. His research interests include artificial intelligence, recommendation system and machine learning.

1 Introduction

With rapid expansion of internet, it is vital to develop more efficient techniques to filter the vast quantities of available information. The collaborative filtering (CF) recommendation algorithm is a widely applied information-filtering technology in recommended systems and is based on user interest. Currently, CF recommendation algorithms are primarily classified into three types: memory-based, model-based and mixed.

Figure 1 shows a typical CF recommendation process, which includes three stages: inputting data, running the CF algorithm and outputting the recommend results. In the input stage, the user ratings of items are inputted into the model. The CF algorithm predicts the unknown ratings of items for users by calculating the similarities between users or items. In the output stage, possible interesting items for users are recommended according to the predicted ratings and, basis which recommendation lists are provided according to ratings from high to low.

Figure 1 A CF recommendation algorithm



A CF recommendation algorithm makes a personalised recommendation according to the user-item rating matrix information; this results in problems such as cold start, low recommendation accuracy and malicious deception. Practically, human tastes and concepts change with time, and human opinions on things differ at different times and on different occasions. In case of the same movie, remarks made differ depending on the user's mood. Therefore, in the design process of recommendation systems, it is necessary to explore how to establish a model to capture these dynamic time effects and give better recommendation results.

Specific to the data sparsity and novelty commonly included in traditional CF recommendation algorithms, a time effect function can be added to the probability matrix factorisation (PMF) algorithm to enable recommendation methods to improve the meeting of user demands. Accordingly, time-based PMF (TPMF) is proposed in this paper. The experimental results using movie datasets indicate that the recommendation accuracy of the TPMF algorithm is higher than traditional CF recommendation algorithms and PMF algorithms.

The remainder of this paper is organised as follows. In Section 4 studies time influence model. Section 5 proposes a time and PMF-based personalised recommendation model. Section 6 introduces the experimental results and the analysis of the related parameters, followed by a conclusion and a description of future work in Section 7.

2 Literature review

The CF algorithm primarily relies on user-item rating matrix information for its recommendations, and the rating data in the matrix are very sparse. To solve the data sparsity problem of the CF algorithm, several studies have proposed various solutions. Resnick et al. (1994) proposed the first automatic CF recommendation system, GroupLens. This system recommends news that is possibly interesting to users. Konstan et al. (1997) enhanced and expanded on the GroupLens system and enabled it to be a complete personalised recommendation system; therefore, establishing a solid foundation for the study and development of CF technology. Breese et al. (1998) conducted a thorough analysis of various CF recommendation algorithms, implemented improvements and achieved certain research results. Deng et al. (2003) proposed an item rating prediction-based CF recommendation algorithm and were the first to calculate the unknown ratings of items by users on the basis of similarities between items. Then, a new similarity calculation method was adopted to calculate neighbour sets of target users, significantly improving the recommendation quality of the system. Xue et al. (2005) proposed a cluster-based method in 2005 and classified users into k types according to the user attribute and cluster number; however, k is not easy to determine. The type closest to the target users was determined as the neighbour of all users in this type. CF technology was then used to calculate the similarity between the target user and its neighbour. This similarity was used to recommend items that were possibly interesting to the users. Iwata et al. (2007) adopted the maximum entropy principle to the CF algorithm to recommend the largest number of possible interesting items. Park and Pennock (2007) integrated a high-efficiency search tool into the CF algorithm to improve the recommendation effects of the system and the user satisfaction. Su et al. (2008) adopted Bayesian classification prediction, mean filling, linear regression prediction and other methods to fill in the original rating matrix and conducted comparison studies of the accuracies of various methods. Kaleli (2014) comprehensively rated the uncertainty difference and the rating similarity for determining the nearest neighbouring set of current users; then they translated it into a knapsack problem.

Time is important contextual information for recommendation systems, and user interests change with time. Adomavicius and Tuzhilin (2001) proposed a multi-dimensional recommendation model, in which space, time and various scene information were integrated into their recommendation process, resulting in better recommendation effects. Sugiyama et al. (2004) proposed a personalised web search engine. In this system, the user configuration files change with time and the system changes the user behaviours via a fixed time delay window; this improves the accuracy of the user search results. Zhao et al. (2005) proposed a time decay function that processes time series data according to each user and item cluster. Using this function, the corresponding time weight of different resources can be calculated. Ding and Li (2005) added a time weight index to a similarity-based CF algorithm. In the rating prediction stage, the ratings of items by users decreased with dynamic time changes. Koren (2010) proposed the TimeSVD++ algorithm, which adds time information to the feature vectors of users and items, improving the recommendation accuracy. In e-commerce, the user purchasing time and item starting time are taken into consideration, which can improve the recommendation accuracy. Lee et al. (2009) proposed a user-rating method that integrates the item release time, user purchasing time or both. In a dynamically changing e-commerce environment, this temporary information can improve the recommendation accuracy of the system. Hong et al. (2012) developed an e-commerce recommendation algorithm that recommends different products according to age and user interest. Zhang and Liu (2015b) proposed a personalised recommendation algorithm integrating a trust relationship and time sequence; this algorithm can effectively improve the recommendation accuracy. In other studies (Zhang and Liu, 2015a; Zhang and Wang, 2017; Qiu et al., 2008; Shao et al., 2007; Hu et al., 2019; Qu and Tao, 2019; Zhang et al., 2017), context-based social networking recommendation algorithms have been proposed; such algorithms provide recommendations for users according to the user's geographical position, time of information and the social relationships of mobile users. Luo et al. (2015) introduced a new probabilistic sequence personalised recommendation model, which uses time properties and dynamic information, to trace interests and preferences of a user, resulting in improving the attraction of the products proposed to that user.

In the above studies, the recommendation results failed to satisfy the user demands. Matrix factorisation technology can be used to solve the problem of data sparsity in a rating matrix; the basic concept is to use a low-rank matrix to approximate the original rating matrix. The goal is to minimise the squared error between the prediction matrix and the original rating matrix. This approach requires a large amount of calculation time. A PMF recommendation algorithm adopts gradient descent, and in the solution procedure, the error descending rate becomes progressively lower. Therefore, the number of iterations and the training time constantly increase. In summary, to the best of our knowledge, none of the previous studies have unified the complementary advantages of a local user and an item neighbourhood, time information and global matrix factorisation models together in CF in a unified probabilistic model.

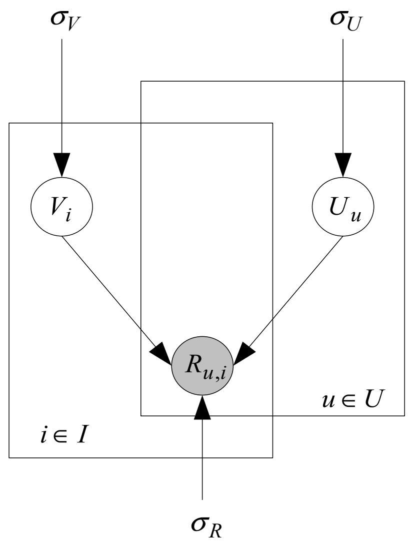
In consideration of the above problems, this paper adopts matrix factorisation technology to complete the rating matrix and decrease the dimension of a sparse rating matrix using the descending dimension method. To guarantee the novelty and effectiveness of the recommended items, this paper proposes a time and PMF-based personalised recommendation algorithm, TPMF. In the experimental section, the influences of the feature vector dimensionality of the users and items on the recommendation accuracy and efficiency are first analysed. Then, experiments are conducted to determine the optimal parameters of the PMF algorithm. Finally, time is added to the PMF algorithm and the influences of time on the recommendation algorithm are analysed. Accordingly, the superiority of the TPMF algorithm is experimentally verified.

The purpose of this study is to present a methodology and specific techniques for modelling time-drifting user preferences with respect to recommendation systems. The proposed approaches are extensively applied on analysed movie rating datasets, enabling us to firmly compare our methods to those recently reported. We show that, by incorporating temporal information, we are able to achieve the best results reported so far, indicating the significance of uncovering temporal effects.

3 Basic PMF model

The PMF model (Mnih and Salakhutdinov, 2007) is depicted in Figure 2. Suppose that the number of users and items in the system are N and M, respectively. $U = \{u_1, u_2, ..., u_N\}$ represents the user set, $I = \{i_1, i_2, ..., i_M\}$ represents the item set and $R_{u,i}$ represents the rating of item i by user u. For a given rating matrix $R = [R_{u,i}]_{N \times M}$, $U \in R^{K \times N}$ and $V \in R^{K \times M}$ represent the feature matrices of the potential users and the potential items, respectively. U_u and V_i represent D-dimensional column vectors, where U_u represents the latent feature vector of a specific user and V_i represents the latent feature vector of a specific item (Zhang and Liu, 2014).

Figure 2 A graphical representation of a PMF algorithm



The feature vectors of the users and products are both supposed to obey Gaussian prior distributions with means of 0.

$$p\left(U \middle| \sigma_U^2\right) = \prod_{u=1}^{N} N\left(U_u \middle| 0, \sigma_U^2 I\right) \tag{1}$$

$$p\left(V \middle| \sigma_V^2\right) = \prod_{i=1}^{M} N\left(V_i \,|\, |0, \sigma_V^2 I\right) \tag{2}$$

The conditional probabilities of the current rating data are defined as follows:

$$p\left(R \middle| U, V, \sigma_R^2\right) = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[N\left(R_{u,i} \middle| g\left(U_u^T V_i\right), \sigma_R^2\right)\right]^{I_{u,i}^R} \tag{3}$$

Here, $N(x \mid \mu, \sigma_R^2)$ represents a normal distribution with a mean of μ and a variance of σ_R^2 and $I_{u,i}^R$ represents an indicator function. If the user u rates i, $I_{u,i}^R$ is 1, otherwise it is 0. $g(x) = 1 / (1 + \exp(-x))$ is a logistic regression function used to limit the predicted value $(U_u^T V_i)$ within the range [0, 1].

Using Bayes inference, the posterior probabilities of the feature vectors U and V can be expressed as follows:

$$\begin{aligned} & p\left(U, V \middle| R, \sigma_R^2, \sigma_U^2, \sigma_V^2\right) \\ & \propto p\left(R \middle| U, V, \sigma_R^2\right) p\left(U \middle| \sigma_U^2\right) p\left(V \middle| \sigma_V^2\right) \\ & = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[N\left(R_{u,i} \middle| g\left(U_u^T V_i\right), \sigma_R^2\right)\right]^{I_{u,i}^R} \\ & \times \prod_{u=1}^{N} N\left(U_u \middle| 0, \sigma_U^2 I\right) \\ & \times \prod_{i=1}^{M} N\left(V_i \middle| 0, \sigma_V^2 I\right) \end{aligned} \tag{4}$$

The posteriori probability, equation (4), is taken as the objective function to solve for the maximum. Taking the logarithm of the predictor equation (4), we obtain equation (5).

$$\begin{aligned} & \ln p\left(U, V \middle| R, \sigma^2, \sigma_V^2, \sigma_U^2\right) \\ & = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \left(R_{ij} - U_i^T V_j\right)^2 \\ & -\frac{1}{2\sigma_U^2} \sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^{M} V_j^T V_j \\ & -\frac{1}{2}\left(\left(\sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}\right) \cdot \ln \sigma^2 + N \cdot D \cdot \ln \sigma_U^2 \right. \\ & \left. + M \cdot D \cdot \ln \sigma_V^2 \right) + C \end{aligned} \tag{5}$$

Here, C represents a constant irrespective of the parameter. Maximising equation (5) can be seen as a non-constrained optimisation problem, and minimising equation (6) is equivalent to maximising equation (5).

$$\begin{aligned} E = & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \left(R_{ij} - U_i^T V_j\right)^2 \\ & + \frac{\lambda_U}{2} \sum_{i=1}^{N} \|U_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^{M} \|V_j\|_{Fro}^2 \end{aligned} \tag{6}$$

Here, $\lambda_U = \sigma^2 / \sigma_U^2$, $\lambda_V = \sigma^2 / \sigma_V^2$ and $\| \cdot \|_{Fro}^2$ denotes the Fresenius norm. Equation (6) is the objective function and its local minimum can be obtained via the gradient descent of U and V.

4 Time influence model

Traditional CF algorithms treat all items equally and fail to consider that the interests of users in items will change with time, which may lead to low accuracy. Because user interests do change with time, adding time to the predicted rating for items will increase the accuracy of the algorithm.

4.1 Calculation of the time weight

In general, the earlier the interest, the lower its importance. This paper adopts an exponential time function to express the influence of time on user interest and preference, which emphasises the ratio of a user's new interests and preferences and reduces the influence of past preferences. The time function is shown in equation (7).

$$f\left(t_{u,i}\right) = 1 \Big/ \left(1 + \exp\left(t_{u,i}\right)\right) \tag{7}$$

$$t_{u,i} = T_{u,i} - T_s \tag{8}$$

Here, $t_{u,i}$ represents the time interval of the ratings of item i by user u, $T_{u,i}$ represents the rating time of item i by user u and T_s represents a past fixed time, which is the base point of the interval time calculation. $f(t_{u,i})$ falls in the range of (0, 1) and represents the time weight of the user interest. A closer rating time indicates a greater value.

4.2 Time effect-based CF model

Traditional CF models predict the rating of an item i by the current user u according to the rating similarities between users or items. The calculation equation is shown in equation (9).

$$P_{u,i} = \bar{R}_u + \frac{\sum_{a=1}^{N} \left(R_{a,i} - \bar{R}_a\right) \cdot sim(u, a)}{\sum_{a=1}^{N} |sim(u, a)|} \tag{9}$$

Here, $sim(u, a)$ represents the rating similarity between the users u and a. Equation (10) calculates the similarity between users u and a using the Pearson correlation coefficient.

$$sim(u, a) = \frac{\sum_{i \in I_{u,a}} \left(R_{u,i} - \bar{R}_u\right)\left(R_{a,i} - \bar{R}_a\right)}{\sqrt{\sum_{i \in I_{u,a}} \left(R_{u,i} - \bar{R}_u\right)^2} \sqrt{\sum_{i \in I_{u,a}} \left(R_{a,i} - \bar{R}_a\right)^2}} \tag{10}$$

Here, $I_{u,a}$ represents the item set rated by users u and a, $R_{u,i}$ and $R_{a,i}$ represent the ratings of item i by the users u and a, respectively, and $\bar{R}_u$ and $\bar{R}_a$ represent the mean ratings of all items by users u and a, respectively.

Liu et al. proposed a time weighted and user feature-based CF algorithm (Liu et al., 2012). They added time effects to traditional CF algorithms to improve the prediction accuracy. The rating prediction method is shown in equation (11).

$$P_{u,i} = \overline{R}_u + \frac{\sum_{a=1}^{n}\left(R_{a,i} - \overline{R}_a\right)\cdot sim(u,a)\cdot f\left(t_{a,i}\right)}{\sum_{a=1}^{n}\left|sim(u,a)\right|\cdot f\left(t_{a,i}\right)} \tag{11}$$

Here, $f(t_{a,i})$ represents the time function and the calculation equation is equation (7).

5 Time information and PMF-based personalised recommendation model

5.1 Time and matrix factorisation recommendation model

According to equation (4), the PMF algorithm only predicts ratings according to the user-item rating matrix and then learns the corresponding feature vector. However, this model does not consider changes in the users' interests in items over time. Accordingly, this paper improves the evaluation model and proposes a time and matrix factorisation recommendation model, TPMF.

Equation (12) calculates the predicted ratings of item i by user u according to the latent feature vectors of the users and items.

$$\hat{r}_{ui} = p_u \cdot q_i^T = \sum_{k=1}^{K} p_{uk}\cdot q_{ik} \tag{12}$$

Equation (12) does not consider the influences of time on the user ratings. After time effects are considered, equation (12) is rewritten, obtaining the predicted rating calculation equation (13) of item i by user u at a time t_{ui}.

$$\hat{r}_{ui} = p_u\left(t_{u,i}\right)\cdot q_i^T = \sum_{k=1}^{K} p_{uk}\left(t_{u,i}\right)\cdot q_{ik} \tag{13}$$

Here, $p_{uk}(t)$ can be obtained from equation (14) below.

$$p_{uk}(t) = p_{uk} + \alpha_{uk}\cdot f(t) \tag{14}$$

In the above equation, α_{uk} represents the regulation parameters and can be obtained by minimising the objective function via gradient descent. $f(t)$ can be obtained from equation (7). Equation (14) is substituted into equation (13), obtaining the predicted rating calculation shown in equation (15).

$$\begin{aligned}\hat{r}_{ui} &= p_u\left(t_{u,i}\right)\cdot q_i^T = \sum_{k=1}^{K} p_{uk}\left(t_{u,i}\right)\cdot q_{ik} \\ &= \sum_{k=1}^{K}\left(p_{uk} + \alpha_{uk}\cdot f\left(t_{u,i}\right)\right)\cdot q_{ik}\end{aligned} \tag{15}$$

Equation (15) is substituted into equation (6), obtaining the predicted rating calculation shown in equation (16).

$$\begin{aligned}E &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{ui}\left(r_{ui} - p_u\left(t_{u,i}\right)\cdot q_i^T\right)^2 \\ &\quad + \lambda\cdot\left(\left\|p_u\right\|_2^2 + \left\|q_i\right\|_2^2 + \left\|\alpha_u\right\|_2^2\right) \\ &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{ui}\left(r_{ui} - \left(p_u + \alpha_u\cdot f\left(t_{u,i}\right)\right)\cdot q_i^T\right)^2 \\ &\quad + \lambda\cdot\left(\left\|p_u\right\|_2^2 + \left\|q_i\right\|_2^2 + \left\|\alpha_u\right\|_2^2\right)\end{aligned} \tag{16}$$

5.2 Gradient descent algorithm

Gradient descent is performed on the objective function E, obtaining the parameters p_u, α_u and q_i. The specific calculation equations are shown as follows:

$$p_u = p_u + \eta\cdot\left(e_{ui}\cdot q_i - \lambda\cdot p_u\right) \tag{17}$$

$$\alpha_u = \alpha_u + \eta\cdot\left(e_{ui}\cdot q_i\cdot f(t) - \lambda\cdot\alpha_u\right) \tag{18}$$

$$q_i = q_i + \eta\cdot\left(e_{ui}\cdot\left(p_u + \alpha_u\cdot f\left(t_{u,i}\right)\right) - \lambda\cdot q_i\right) \tag{19}$$

Here, $e_{ui} = r_{ui} - \hat{r}_{ui} = r_{ui} - p_u(t_{u,i})\cdot q_i^T$, p_u and the initial values of α_u and q_i are both random numbers. The recommendation algorithm aims to minimise the difference. For the penalty factor λ and the learning rate η, an initial value is usually assigned to λ according to experience and then η is adjusted. Then, η is fixed and λ is adjusted. Finally, according to the experimental effects, the optimal values of λ and η are chosen.

5.3 Analysis of the algorithm performance

The time complexity of the matrix factorisation algorithm primarily arises from gradient descent, and its calculation costs result from the objective function E and the corresponding gradient descent equation. The matrices U and V are sparse, and the time complexity of the objective function in equation (16) is $O(D\cdot n + D\cdot m)$, where n and m represent the number of non-zero elements in the matrices U and V, respectively. Therefore, the time complexity of each iteration can be expressed as $O(D\cdot n + D\cdot m)$. Namely, the time complexity of the algorithm increases linearly with the increasing amount of observation data in the sparse matrix. Therefore, the expansibility of the algorithm is good and it can be applied in recommendation systems under the context of large-scale data.

6 Experimental results and analysis

6.1 Datasets

This experiment includes two datasets, Netflix and MovieLens. The Netflix dataset is a movie rating dataset released by Netflix in October 2006. This dataset contains over 100 million ratings from 31 December 1999, to

31 December 2005. These ratings are for 17,770 movies by 480,000 anonymous users, and each rating has a rating date. In this dataset, each movie has 5,600 ratings and each user rates 208 movies on average. The Netflix dataset was released for the Netflix prize competition with the goal of improving the recommendation system of the company. In the competition, the recommendation effects needed to be improved by 10% compared to the Cinematch recommendation system. Compared to Netflix recommendation system, the Cinematch system has an RMSE of 0.9514 on the test dataset. The MovieLens dataset was collected by the GroupLens research group at the University of Minnesota and is one of the most successfully applied datasets. The MovieLens evaluation information of movies has obtained wide acceptance and has been widely used in simulation tests of personalised recommendation algorithms.

6.2 Metrics

The accuracy of the predicted ratings represents the difference between the actual user ratings and the ratings predicted by the algorithm. This paper primarily uses the two metrics below.

Mean absolute error (*MAE*):

$$MAE = \frac{1}{|E^P|} \sum_{(u,i)\in E^P} |r_{ui} - \hat{r}_{ui}| \tag{20}$$

Here, E^P represents the test dataset and $|E^P|$ represents the number of items rated by users in the test dataset.

Root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{|E^P|} \sum_{(u,i)\in E^P} (r_{ui} - \hat{r}_{ui})^2} \tag{21}$$

For the prediction of new items, lower values of these two metrics indicate higher prediction accuracies.

6.3 Experimental scheme

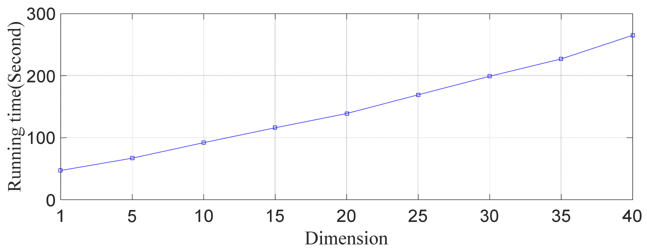
In real systems, determining the feature vector dimensionality of the users and items in a PMF algorithm is a key influence factor in the recommendation accuracy. However, researchers usually assign values according to experience. Therefore, determining an optimal value has always been a research problem. The PMF algorithm has a longer program running time and lower recommendation efficiency with the increasing feature vector dimensionality of the users and items. How to reduce the feature vector dimensionality and improve the operation efficiency of the algorithm while guaranteeing the recommendation accuracy is a very important research problem. The reason for this is that the acceptable operation time of users is short.

This paper conducted four sets of experiments.

1. By changing the feature vector dimensionality D, its influences on the running time of the PMF algorithm were analysed.
2. By changing the feature vector dimensionality D, its influences on the RMSE were analysed.
3. Using the Netflix dataset, the recommendation accuracies of the TPMF algorithm, the baseline Netflix system recommendation algorithm, the SVD algorithm and the PMF algorithm were compared.
4. Using the MovieLens dataset, the recommendation accuracies of the TPMF algorithm and a traditional CF recommendation algorithm were compared.

Experiment 1 was used to study the influences of the dimensionality D on the running time. Experiment 2 was used to study the influences of the dimensionality D on the recommendation accuracy. Experiments 3 and 4 were used to test the recommendation accuracy of the TPMF algorithm.

Figure 3 The effect of dimensionality on the running time of the PMF algorithm (see online version for colours)



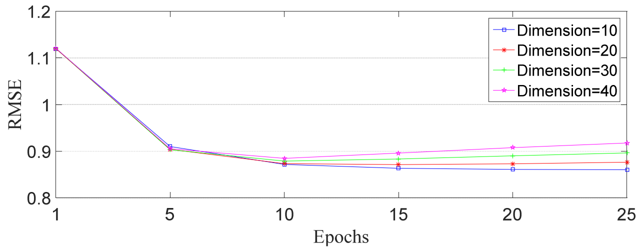
6.4 Influences of dimensionality on the running time of the PMF algorithm

Experiment 1 was designed to verify the influences of the feature vector dimensionality of the users and items on the running time, and the experimental results for which are shown in Figure 3. It can be seen that, with increases in the feature vector dimensionality, the execution time of the PMF algorithm becomes longer and the operational efficiency decreases greatly.

6.5 Influences of dimensionality on the recommendation accuracy

Experiment 2 was designed to analyse the influences of changing the feature vector dimensionality D on the RMSE, and the experimental results are shown in Figure 4. It can be seen that, with increasing feature vector dimensionality, when the iteration epoch is greater than 10, the RMSE gradually increases. Therefore, when the feature vector dimensionality of the matrix increases to a certain level, the recommendation accuracy decreases with the addition of noise. Experiment 2 indicates that the optimal feature vector dimensionality is 10. Therefore, experiments 3 and 4 adopt a feature vector dimensionality of D = 10 as the initial value.

Figure 4 The effect of dimensionality on the RMSE (see online version for colours)

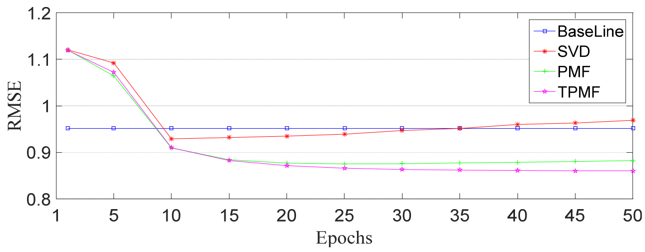


6.6 *Comparison of the recommendation accuracy*

To test the recommendation accuracy of the TPMF algorithm in this paper, TPMF was verified using the Netflix and MovieLens datasets in experiments 3 and 4.

First, using the Netflix dataset, the TPMF algorithm, the baseline Netflix system recommendation algorithm, the SVD algorithm and the PMF algorithm were compared using the RMSE metric. The Netflix system recommendation algorithm had an $RMSE = 0.9514$. PMF had the mean RMSE when $D = 5, 10, 20, 30, 40$. The comparison results are shown in Figure 5. The x-axis represents the iteration epochs, and the y-axis represents the RMSE.

Figure 5 Performance comparisons of the four methods (see online version for colours)



It can be seen in Figure 5 that the SVD algorithm experiences serious overfitting. When *Epochs* is greater than 10, the SVD algorithm starts overfitting. The recommendation accuracy of the TPMF algorithm is 7.6% higher than that of the SVD algorithm, approximately 9.7% higher than that of the baseline Netflix benchmark algorithm and approximately 1.9% higher than that of the PMF algorithm. Therefore, in sparse matrices and unbalanced datasets, the TPMF algorithm performs better than the other recommendation algorithms with respect to prediction accuracy.

Second, using the MovieLens dataset, the TPMF algorithm and the traditional CF algorithm were compared using the MAE metrics. The CF algorithm recommends similar items to target users according to the similarities between items. The similarities between users or items can be obtained using equation (10). The experiment in this section randomly selects three users: users 3, 8 and 11. The number of neighbours is 5 at first and then increases to 25 with an interval of 5. The experimental results are shown in Figures 6, 7 and 8.

Figure 6 Performance comparisons for user 3 (see online version for colours)

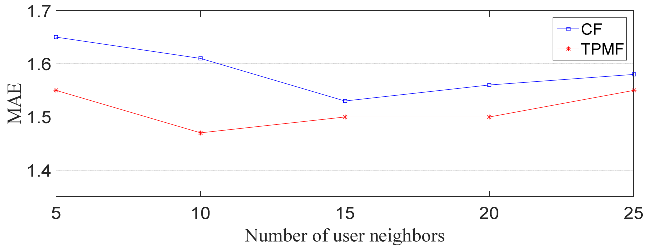


Figure 7 Performance comparisons for user 8 (see online version for colours)

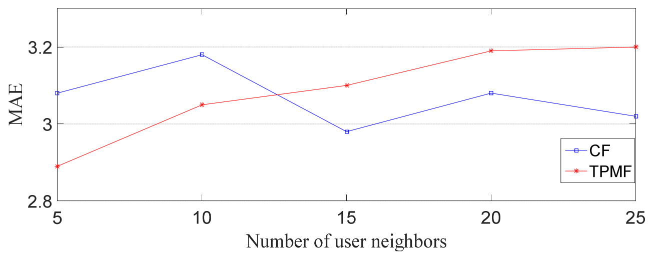
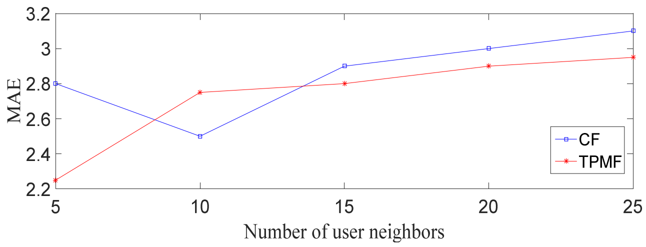


Figure 8 Performance comparisons for user 11 (see online version for colours)



Figures 6, 7 and 8 indicate that, compared to the traditional CF algorithm, the improved matrix factorisation algorithm TPMF has a much better recommendation quality, regardless of the user. In Figure 6, when there are ten neighbours, the MAE value of the TPMF algorithm reaches a minimum and indicates the highest recommendation quality. Correspondingly, in Figures 7 and 8, when the number of neighbours are 15 and 5, respectively, the MAE values of the TPMF algorithm reach a minimum and indicate the best recommendation effects.

Therefore, PMF technology, which supplements the rating matrix and reduces the dimensionality of a sparse matrix via the descending dimension method, can avoid the condition in which it is difficult to look up neighbouring users when the user rating information is sparse. This paper adjusts the timeliness and importance of the user ratings by introducing a time function. Under the condition of a changeable rating weight, the rating similarity between users is calculated, which effectively improves the recommendation accuracy of the algorithm.

7 Conclusions

Following the drawbacks of traditional CF recommendation algorithms, this paper thoroughly explored CF algorithm and matrix factorisation technology. As an important implementation of the latent factor model, matrix factorisation technology has been well applied to recommendation systems. If product deviation and user deviation effects with time are added to the basic matrix factorisation model, the influences of product evaluation deviation, user preference deviation and the daily mood fluctuations of users on scores can be captured. Accordingly, this paper proposed a time and PMF-based personalised recommendation algorithm. Using experiments, the influences of the feature vector dimensionality of the users and items on the recommendation accuracy and efficiency were analysed; thus, verifying the consistency between the training sets and datasets in the recommendation results. Further, the optimal PMF parameters were determined and the influences of the feature vector dimensionality on the recommendation accuracy and efficiency were analysed. The experimental results on real datasets indicate that, after time effects are considered, the prediction deviation is decreased significantly and the prediction accuracy is improved greatly. The TPMF algorithm is a better method under data sparsity and its time complexity is low. The CF recommendation algorithm relies on user history data, including user interests, preferences and purchasing habits; hence, such data may involve user privacy and result in many hidden dangers and confusion for users. Therefore, how to provide satisfactory personalised recommendation to users while effectively protecting users' privacy is a future research direction.

Acknowledgements

This paper is made possible thanks to the generous support from the National Social Science Foundation of China (16BGL137), the Key Research and Development Program of Shandong Province (2018GGX106006, 2019GGX101068, 2019GGX101004), Jinan Science and Technology Project (201704065), Science and Technology Program Project of Shandong Colleges and Universities (J17KA070), Doctoral Foundation of Shandong Jianzhu University (XNBS1523，X19044Z), Shandong Education Department Teaching Reform Project (Z2016M016，Z2016M014，M2018X197), China Scholarship Council(CSC201809995006).

References

Adomavicius, G. and Tuzhilin, A. (2001) 'Multidimensional recommender systems: a data warehousing approach', *Electronic Commerce*, pp.180–192, Springer, Berlin Heidelberg.

Breese, J.S., Heckerman, D. and Kadie, C. (1998) 'Empirical analysis of predictive algorithms for collaborative filtering', *Uncertainty in artificial intelligence*, July, Vol. 98, No. 7, pp.43–52.

Deng, A., Zhu, Y. and Shi, B. (2003) 'A collaborative filtering recommendation algorithm based on item rating prediction', *Journal of Software*, September, Vol. 14, No. 9, pp.1621–1628.

Ding, Y. and Li, X. (2005) 'Time weight collaborative filtering', in *Proc. of the 14th ACM International Conference on Information and Knowledge Management*, ACM, pp.485–492.

Hong, W., Li, L. and Li, T. (2012) 'Product recommendation with temporal dynamics', *Expert Systems with Applications*, August, Vol. 39, No. 16, pp.12398–12406.

Hu, B., Wang, H. and Zheng, Y. (2019) 'Sign prediction and community detection in directed signed networks based on random walk theory', *International Journal of Embedded Systems*, Vol. 11, No. 2, pp.200–209.

Iwata, T., Saito, K. and Yamada, T. (2007) 'Modeling user behavior in recommender systems based on maximum entropy', in *Proc. of the 16th International Conference on World Wide Web*, ACM, pp.1281–1282.

Kaleli, C. (2014) 'An entropy-based neighbor selection approach for collaborative filtering', *Knowledge Based Systems*, March, Vol. 56, No. 3, pp.273–280.

Konstan, J.A., Miller, B.N., Maltz, D. et al. (1997) 'GroupLens: applying collaborative filtering to Usenet news', *Communications of the ACM*, March, Vol. 40, No. 3, pp.77–87.

Koren, Y. (2010) 'Collaborative filtering with temporal dynamics', *Communications of the ACM*, April, Vol. 53, No. 4, pp.89–97.

Lee, T.Q., Park, Y. and Park, Y.T. (2009) 'An empirical study on effectiveness of temporal information as implicit ratings', *Expert Systems with Applications*, February, Vol. 36, No. 2, pp.1315–1321.

Liu, D., Peng, D. and Zhang, H. (2012) 'Collaborative filtering algorithm based on time weight and user's feature', *Journal of Wuhan University of Technology*, May, Vol. 34, No. 5, pp.144–147.

Luo, C., Cai, X. and Chowdhury, N. (2015) 'Probabilistic temporal bilinear model for temporal dynamic recommender systems', in *2015 International Joint Conference on Proc. of Neural Networks (IJCNN)*, IEEE, pp.1–8.

Mnih, A. and Salakhutdinov, R. (2007) 'Probabilistic matrix factorization', *Advances in Neural Information Processing Systems*, February, Vol. 20, No. 2, pp.1257–1264.

Park, S.T. and Pennock, D.M. (2007) 'Applying collaborative filtering techniques to movie search for better ranking and browsing', in *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.550–559.

Qiu, H., Qiu, M. and Lu, Z. (2020) 'Selective encryption on ECG data in body sensor network based on supervised machine learning', *Information Fusion*, Vol. 55, pp.59–67.

Qiu, M., Sha, H.M., Liu, M. et al. (2008) 'Energy minimization with loop fusion and multi-functional-unit scheduling for multidimensional DSP', *Journal of Parallel and Distributed Computing*, Vol. 68, No. 4, pp.443–455.

Qu, B. and Tao, X. (2019) 'An efficient privacy-preserving friendship based recommendation system', *International Journal of Embedded Systems*, Vol. 11, No. 4, pp.516–525.

Resnick, P., Iacovou, N., Suchak, M. et al. (1994) 'GroupLens: an open architecture for collaborative filtering of netnews', in *Proc. of the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, USA, pp.175–186.

Shao, Z., Wang, M., Chen, Y. et al. (2007) 'Real-time dynamic voltage loop scheduling for multi-core embedded systems', *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 54, No. 5, pp.445–449.

Su, X.Y., Khoshgoftaar, T.M. and Greiner, R. (2008) 'A mixture imputation-boosted collaborative filter', in *Proc. of the 21st International Florida Artificial Intelligence Research Society Conference*, Coconut Grove, USA, pp.312–316.

Sugiyama, K., Hatano, K. and Yoshikawa, M. (2004) 'Adaptive web search based on user profile constructed without any effort from users', in *Proc. of the 13th International Conference on World Wide Web*, ACM, pp.675–684.

Xue, G.R., Lin, C., Yang, Q. et al. (2005) 'Scalable collaborative filtering using cluster-based smoothing', in *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp.114–121.

Zhang, D., Shou, Y. and Xu, J. (2017) 'An improved parallel K-means algorithm based on MapReduce', *International Journal of Embedded Systems*, Vol. 9, No. 3, pp.275–282.

Zhang, Z. and Liu, H. (2014) 'Application and research of improved probability matrix factorization techniques in collaborative filtering', *Int. J. Control. Autom.*, August, Vol. 7, No. 8, pp.79–92.

Zhang, Z. and Liu, H. (2015a) 'Research on context-awareness mobile SNS recommendation algorithm', *Pattern Recognition and Artificial Intelligence*, May, Vol. 28, No. 5, pp.404–410.

Zhang, Z. and Liu, H. (2015b) 'Social recommendation model combining trust propagation and sequential behaviors', *Applied Intelligence*, March, Vol. 43, No. 3, pp.695–706.

Zhang, Z. and Wang, Y. (2017) 'Research on SNS trust comprehensive evaluation based personalized recommendation algorithm', *Applied Intelligence*, March, Vol. 47, No. 3, pp.659–669.

Zhao, Y., Zhang, C. and Zhang, S. (2005) 'A recent-biased dimension reduction technique for time series data', *Advances in Knowledge Discovery and Data Mining*, pp.751–757, Springer, Berlin Heidelberg.