Noname manuscript No.
(will be inserted by the editor)

Personality-Boosted Matrix Factorization for Recommender Systems

Mazyar Ghezelji · **Chitra Dadkhah**

Received: date / Accepted: date

Abstract Recommender systems are one of the extensively used Knowledge Discovery in Database techniques which have gained a lot of attention recently. These systems have been applied in many internet-based communities and businesses to make personalized recommendations and eventually gain a higher profit. The core entity in these systems is ratings from users to items but there are many auxiliary pieces of information that can be used to get better performance. The personality of users is one of the most useful information that helps the system to produce more accurate and personalized recommendations. It has been proved that the characteristics of a person can directly affect her behavior.In this paper, We exploit the personality of users and propose a mathematical and algorithmic approach to utilize this information. the base model that we use is matrix factorization which is one of the most powerful methods in recommender systems. Experimental results on real-world datasets demonstrate the positive impact of personality information on the matrix factorization technique and also reveals better performance compared to state-of-the-art algorithms.

Keywords Recommender system · Matrix Factorization · Knowledge Discovery

1 Introduction

The rapid growth of e-commerce and social media has led to a great source of information on the web which causes information overload [9]. This information overload creates some difficulties for the users to find their desired items or information. Eventually, this specific problem will decrease user satisfaction and loyalty [8]. To handle this problem recommender systems have been introduced. A recommender system is a special type of Knowledge Discovery in Databases technique that generates personalized recommendations by analyzing the patterns of users' behavior and interests [5, 6]. Various kinds of recommender systems

Mazyar Ghezelji
Computer Engineering Faculty, K.N.Toosi University of Technology
E-mail: maz.ghezelji@email.kntu.ac.ir

Chitra Dadkhah
Computer Engineering Faculty, K.N.Toosi University of Technology
E-mail: dadkhah@kntu.ac.ir

have been invented by utilizing three main entities in the system which are user, item and user-item interactions [14].
Three main types of recommender systems are content-based, collaborative filtering (CF), and hybrid recommender systems [2]. In the content-based approach, recommendations are produced based on similarities in users preferences and items attributes. The collaborative filtering approach doesn't use the attributes features, instead, it evaluates the items based on the opinions of other people [3]. Hybrid recommender systems are a combination of different approaches which sometimes gives better results.
Collaborative filtering has gained many researchers and e-retailers' attention due to its attributes: CF does not require domain knowledge, means it can work in systems where gaining attributes of items are difficult or can not be automated [6, 7]. It also can provide a serendipitous recommendation as it does not only recommend items similar to just a specific users history but it takes advantage from a group of users which helps users discover new items [7]. Two major approaches in collaborative filtering are neighborhood-based and matrix factorization approaches [13]. Neighborhood-based uses several similar users to predict the preferences but matrix factorization represents users and items with a set of latent factors to make it directly comparable.
The neighborhood-base approach can find local relations among users and can give a relatively good result when the number of users is not huge and the rating matrix is not so sparse. But as the size of online communities increase and the number of users grows, the accuracy of the CF method decreases and it gets harder to implement it at scale. Matrix factorization method can find overall structure unlike the neighborhood-based approach and has good performance in large datasets [6, 11], it also can be integrated with auxiliaries to mitigate cold start problem and improve the accuracy.
Most of the base recommender systems use ratings from users to items to predict the preferences but there are many auxiliary information like the context of ratings and personality information that could improve this method [10, 12, 13]. Human personality, as the identity of human, affects all aspects of life including behavior in social networks, online shops, etc. The correlation of personality and rating behavior has been widely studied before and the results show that leveraging personality information can help collaborative filtering methods get a better result and alleviate cold-start problem [7, 10].
The previous methods which use personality information, gain this information usually from other data sources, or by mining the contents of items and profiles of users, but these data sources are not always available. If the only data we have access to is the rating matrix, does it mean that we can't tell the personalities? we answer that it is not a limitation. The personality of users affect their rating behavior and could be obtained from it too. The authors in [1] has shown that personality of a user in terms of optimism and pessimism can be obtained from her ratings for items and other users and it can affect her positive and negative relations with other users. Our main contributions in this paper are as follows:

- We define the personality of users from their rating behavior and do some analysis on real-world data with regard to this new concept in order to gain some new information.
- We propose a novel matrix factorization method by utilizing personality information.
- We evaluate models and reveal the advantage of our model through its comparison with other state-of-the-art models.

The remainder of the paper is organized as follows: in section 2 we define the main problem, in section 3 we make a hypothesis to be used in MF and test it on the dataset, in section 4 we discuss our proposed model and in section 5 we evaluate and compare the algorithm to other algorithms.

2 Problem Definition

The main data used in recommender systems is the user-item rating matrix. It shows that how much a user tends an item using some numbers on a scale. The scale and the meaning of ratings depends on the platform, for example in Amazon ratings are between 1 and 5 which 1 means a user totally dislike an item and 5 means a user totally likes an item but in Steam which is a gaming platform, 1 means a user has purchased a game and in case the user plays the game the rate is equal to total hours of playing so the scale is not limited.
The main idea of the recommendation system is to find a set of items that a user will be interested in. Lets assume $U = \{u_1, u_2, \ldots, u_n\}$ is the set of users and $I = \{i_1, i_2, \ldots, i_m\}$ is the set of items, n is the size of users and m is the size of items. We use $G \in R^{n\times m}$ which is the rating matrix. Each entry $G_{u,i}$ of G is a rating from user u for item i. The ratings are in the range of 1 to 5, and 0 means the user has not seen or rated the item.
Some studies have shown that side information like links between users can be used for increasing matrix factorization accuracy but there are a few studies on finding user attributes in rating matrix and using them for matrix factorization. To get the maximum result from our limited data source, we define personality attributes for each user and use them in the optimization process in section 3. Regarding the data we have, we can define our problem as:
Consider the rating matrix G and personality vector $\vec{\hat{P}}$, our goal is to fill the entries in G which are 0 by predicting them.

3 Preliminaries and data analysis

Our proposed model consists of two parts, Matrix factorization, and personality information in terms of optimism and pessimism.

3.1 Personality

Optimism and pessimism are two important characteristics of users that have a huge impact on their ratings. Obviously, if a user's ratings are mostly more than average she is more optimistic and otherwise [1, 16]. As it is stated in [1] suppose O_u is optimism degree of user u, P_u is pessimism degree of user u and r_{ui} is the rate that user u had given to item i. we can calculate O_u as Eq.1.

$$O_u = \frac{|\{r_{ui}|r_{ui} > 3, \bar{r_i} \leq 3\}|}{|\{r_{ui}|r_{ui} \neq 0, \bar{r_i} \leq 3\}|}. \tag{1}$$

Where r_{ui} is the rating that user u have given to item i and $\bar{r_i}$ is the average of all ratings given to item i. This equation indicates that optimism degree of a user depends on the number of high rates that user u had given to less popular items.
Similarly P_u can be calculated using Eq.2:

$$P_u = \frac{|\{r_{ui}|r_{ui} < 3, \bar{r_i} \geq 3\}|}{|\{r_{ui}|r_{ui} \neq 0, \bar{r_i} \geq 3\}|} \tag{2}$$

Table 1 Statistics of Movielens 100k

	rate	#rates of each user	Personality	Optimism	Pessimism
Min	1	20	-0.79	0	0
Max	5	737	1	1	0.79
Mean	3.52	106.04	0.10	0.23	0.12

which indicates that the pessimism degree of a user depends on the number low rates that user u had given to more popular items. Less popular items are items with the average rate of less than 3 and more popular item are items with the average rate of greater than 3. By comparing these equations to our initial intuitions, we can see that they totally comply our intuition for optimism and pessimism.
We define the personality of a user as Eq.3.

$$Personality(u) = O_u - P_u. \tag{3}$$

This term indicates the total propensity of the user to optimism and pessimism which can be a floating point number in the range of -1 and 1. Personality of 1 means totally optimistic and -1 means totally pessimistic.

3.2 Dataset

For evaluation section we use Movielens 100k dataset [4]. Movielens is a dataset derived from a non-commercial web-based recommender system with the same name for movies. Each user can rate each movie from 1 to 5. Furthermore users can attach tags based on the content of the movie so that the accuracy of the recommender will be increased.
This dataset consists of 100,000 ratings from 943 users on 1682 movies. The dataset has been pre-processed so each user has rated at least 20 movies. Demographic information like age, gender and occupation is also stated for each user which we don't use in the algorithm. This dataset has been split into 5 test and train parts. Each training part is 80% and each test part is 20% of the whole dataset which are taken randomly and could be used for 5-fold cross-validation.
The information about the personality of users in the dataset are mentioned in Table 1.

3.3 Hypothesis

There are several relations among users characteristics and the decisions that they tend to make. In the case of social networks and recommender systems, the influence of their characters on actions and ratings is undeniable. After analyzing the Movielens 100k user-item matrix, users ratings, and their personalities, we constructed a hypothesis about the relation between personality and ratings. An important point is that this hypothesis is about all users who have rated more items than a threshold. As mentioned before, this dataset has been already pre-processed, so the minimum threshold that we can have is 20. We define T as the set of all users who have rated more items than the threshold. our hypothesis is stated in Eq.4.

$$\forall_{u \in T} P_u > \bar{P_T} \rightarrow \bar{r_u} > \bar{r_T} \tag{4}$$

Where u indicates user, P_u indicates the personality of user u, $\bar{P_T}$ is the average personality of all users in the T set, $\bar{r_u}$ is the average of ratings given by user u and $\bar{r_T}$ is the average of rating given by all users in T set. In other words, our hypothesis is:
among all users in the set T, for each user u if the personality of user u is higher than the average personality of all users in T, then the average rating of user u is higher than the average rating of all users in T.

3.4 Hypothesis Testing

As mentioned before, for testing this hypothesis we use a two-sample t-test on the Movielens 100k dataset. We use two arrays $\vec{tp}$ and $\vec{fp}$ indicating true positive and false positive respectively. In our experiment, we check every user throughout the dataset. If the personality of user u is greater/less than the average personality of all users and average of all ratings by user u is greater/less than the average of all ratings in rating matrix then we add 1 to $\vec{tp}$ and 0 to $\vec{fp}$. Otherwise, we add 0 to $\vec{tp}$ and 1 to $\vec{fp}$. We define null hypothesis H_0 and alternative hypothesis H_1 as Eq.5 according to $\vec{tp}$ and $\vec{fp}$ vectors.

$$H_0 : \bar{tp} \leq \bar{fp} \qquad H_1 : \bar{tp} > \bar{fp} \tag{5}$$

The result of t-test shows that the null hypothesis is rejected with t-statistics equal to 18.5745 and p-value equal to 3.724477401546685e-71. So we can infer that our hypothesis is true, it means if a user's personality is more than average, then we can say her average rating is also more than average.

4 Personality-Boosted Matrix Factorization

Matrix factorization is a very popular model-based collaborative filtering technique. Its scalability, accuracy, ability to integrate regularizations, and ability to circumvent cold-start problem gives us the confidence to use it as the basic model. The main idea of MF is to decompose the rating matrix into two smaller matrixes which the product of them gives back the rating matrix. In this literature, R is the rating matrix with $m \times n$ dimensions, U is the user matrix with $m \times k$ dimensions and V is items matrix with $n \times k$ dimensions which m is the number of users, n is the number of items and k is the number of latent factors. MF process is as Eq.6.

$$R = U.V^T \tag{6}$$

and therefore finding the U and the V matrix is through solving the optimization problem in Eq.7.

$$\min_{U,V} ||W \odot (R - U \cdot V^\top)||_F^2 + \lambda_1 \cdot ||U||_F^2 + \lambda_2 \cdot ||V||_F^2 \tag{7}$$

where W is the weight matrix, usually $W_{ui} = 1$ if there is a rating from user u for item i otherwise $W_{ui} = 0$, $||.||$ is Frobenius norm and $\odot$ is element-wise product of two matrix . The terms $||U||_F^2$ and $||V||_F^2$ are regularization terms which prevent overfitting. The most well-known method for this optimization problem is stochastic gradient descent which We will discuss it later.
We want to apply some regularization for the purpose of taking advantage of our hypothesis. Suppose that the personality of a user is more than average personality, but her average

rating is less than average rate of all users. In this case we can say the hypothesis is violated. So we add the difference of her average rating and total average rating as a penalty to the optimization function. In another form, the term that we add to the optimization problem to be minimized is stated in Eq.8.

$$\min \sum_{u|P_u>\bar{P}} \max(0, \bar{r}-\bar{r_i})^2 + \sum_{u|\bar{P}>P_u} \max(0, \bar{r_i}-\bar{r})^2 \tag{8}$$

where u indicates every user in the system. The first part is for the case that personality of user is more than average personality and the other part is for the case that personality of user is less than average personality.
Putting all the parts together we can write the optimization problem as Eq.9.

$$\begin{aligned} &\min_{U,V} \|W \odot (R - U \cdot V^\top)\|_F^2 + \lambda_1 \cdot \|U\|_F^2 + \lambda_2 \cdot \|V\|_F^2 \\ &+ \sum_{u|P_u>\bar{P}} \max(0, \bar{r}-\bar{r_i})^2 + \sum_{u|\bar{P}>P_u} \max(0, \bar{r_i}-\bar{r})^2. \end{aligned} \tag{9}$$

Because we use max function, There is no closed-form solution for this problem. Therefore we use gradient descent to get a acceptable local minimum. Gradient descent method uses derivative of optimization function in order to shift the solution towards a better one. Using gradient descent requires nice matrix form equation so we need to rewrite the formulation. First we define new terms for new equation:

- Vector A with m elements where its elements are 1 if $P_i \le \bar{P}$ and $r_i >= \bar{r}$, -1 if $P_i > \bar{P}$ and $r_i < \bar{r}$ and 0 otherwise.
- Vector X with n elements where its elements are equal to $\frac{1}{n}$.
- Vector Y with m elements where its elements are equal to $\bar{r}$.

Using new terms the formulations are as Eq.10.

$$\begin{aligned} &\min_{U,V} \frac{1}{2} \cdot \|W \odot (R - U \cdot V^\top)\|_F^2 + \frac{\lambda_1}{2} \cdot \|U\|_F^2 + \frac{\lambda_2}{2} \cdot \|V\|_F^2 \\ &+ \frac{\lambda_3}{2} \cdot (A^\top \cdot (U \cdot V^\top \cdot X - Y))^2. \end{aligned} \tag{10}$$

Now we take derivative of optimization function with respect to U and V for updating step in gradient descent process. Showing the optimization function by F, derivative of F with respect to U is stated in Eq.11.

$$\begin{aligned} &\frac{\partial F}{\partial U} = \lambda_1 \cdot U - (W \odot (R - U \cdot V^\top) \odot W) \cdot V \\ &+ \lambda_3 \cdot (X^\top \cdot V \cdot U^\top + (-Y)^\top) \cdot A \cdot A \cdot (X^\top \cdot V) \end{aligned} \tag{11}$$

and derivative of F with respect to V is stated in 12.

$$\begin{aligned} &\frac{\partial F}{\partial V} = \lambda_2 \cdot V - (W^\top \odot (R^\top - V \cdot U^\top) \odot W^\top) \cdot U \\ &+ \lambda_3 \cdot A^\top \cdot (U \cdot V^\top \cdot X - Y) \cdot X \cdot (A^\top \cdot U). \end{aligned} \tag{12}$$

Notice that element-wise product of W with W is not necessary as it is a matrix of 1s and 0s, but we have written it to keep the generalization of the solution.

The complete algorithm is written in algorithm 1. In line 1 we calculate W, X and Y is it was discussed before. In line 2 we calculate Personality of all users and store it in Personality vector. In line 3 and 4 we initialize user matrix, item matrix and counter for learning loop. From line 5 to 12, for each iteration of the learning loop we calculate vector A from the predicted rates and update user matrix U and item matrix V. Finally the algorithm returns user matrix and item matrix. Predicted rating matrix is $U \cdot V^\top$.

Algorithm 1: Personality-Boosted Matrix Factorization

Input: the rating matrix **G**, regularization coefficients $\lambda_1, \lambda_2, \lambda_3$, number of latent factors **K**,number of iterations **i**, learning rate α.
Output: **U** and **V**

1 Calculate W ,X,Y ;
2 Calculate Personalities;
3 Initialize U,V with random elements;
4 Define $l = 0$
5 **while** l != i **do**
6   Calculate A ;
7   Calculate $\frac{\partial F}{\partial U}$;
8   Calculate $\frac{\partial F}{\partial V}$;
9   Update $U \leftarrow U - \alpha \frac{\partial F}{\partial U}$;
10   Update $V \leftarrow V - \alpha \frac{\partial F}{\partial V}$;
11   Update $l \leftarrow l + 1$;
12 **end**
13 Return U,V ;

5 Evaluations

In this section, we are going to assess our algorithm's performance on real data and also compare it to the other powerful state-of-the-art algorithms.

5.1 Evaluation Metrics

The procedure that we use in this section for evaluation is cross-validation. To use cross-validation we should first split the dataset into some equal parts then each time set one part as test data and the others as train data. We repeat this procedure for all parts then the final results are the average result of all tests.
To compare the different methods and demonstrate the quality of their predictions we use MAE, MSE and RMSE metrics. MAE is the mean absolute error and is defined as Eq.13.

$$MAE = \frac{\sum_{i \in R} |r_i - \hat{r_i}|}{N} \tag{13}$$

MSE is the mean squared error and is defined as Eq.14

$$MSE = \frac{\sum_{i \in R} (r_i - \hat{r_i})^2}{N} \tag{14}$$

RMSE is the root of MSE as Eq.15.

$$RMSE = \sqrt{\frac{\sum_{i \in R}(r_i - \hat{r_i})^2}{N}} \tag{15}$$

Where r_i is the actual rate, $\hat{r_i}$ is the predicted rate and N is the total number of ratings.

5.2 Comparisons

the algorithms that we have selected for comparison are:

- MF [6, 8]
  Matrix Factorization is a well-known model-based colaborative filtering algorithm. it is the base for many powerful algorithms because of its flexibility and scalability.
- KNN [8]
  K-Nearest Neighbor algorithm is another model-based colaborative filtering that is based on clustering.
- Co-Clustering [15]
  A dynamic colaborative filtering based approach for recommender systems.
- NMF [14]
  Non-negative Matrix Factorization is a type collaborative filtering based on matrix factorization technique which both users and items are kept positive.
- PBMF
  Personality-Boosted Matrix Factorization is our proposed algorithm. it was discussed in section 4.

The similarity measure in KNN method is cosine similarity. We set maximum number of k equal to 40 and the minimum number of k equal to 1. The reason for using maximum and minimum for k is that some users have less neighbors than maximum k and further more we can't set users with negative similarity as friends.
Nmber of factors used in NMF algorithm is 15, regularization terms for users and items are both 0.06 and we have optimized it using stochastic gradient descent for 50 epochs.
For the co-clustering algorithm, the number of user and item clusters are both 3 and the number of iterations optimization loop is 20. As the result of co-clustering and NMF algorithms are dependent on initial state we repeat the experiment with 5 different random initial state and take the average as the final result and for the MF and PBMF the initial state is matrices with all elements of 1.
The iteration of optimization loop in MF and PBMF are both 80, λ_1, and λ_2 in both methods are 0.05. Number of latent factors in MF is 2 and in PBMF is 3. λ_3 in PBMF is 4. we will discuss appropriate λ_1, λ_2, λ_3 and k later in this section.
For better comparison we divide the algorithms into two groups of matrix factorization based algorithms and the algorithms which are not based on matrix factorization. The detailed comparison of algorithms are shown in table 2 and we can see the visualized difference from figure 1.

- Form table 2 and figure 1 we can observe that matrix factorization based algorithms have a relatively better performance compared to other types of collaborative filtering methods. among the non-MF based algorithms, the co-clustering algorithm gives 0.0136 reduction over KNN in terms of RMSE and 0.0179 in terms of MAE.

Table 2 MAE and RMSE of all algorithms

Algorithm	RMSE	MAE
K Nearest Neighbor	0.9801	0.7744
Co-Clustering	0.9665	0.7565
Non-negative Matrix Factorization	0.9669	0.7598
Matrix Factorization	0.9554	0.7581
Personality-Boosted Matrix Factorization	0.9468	0.7493

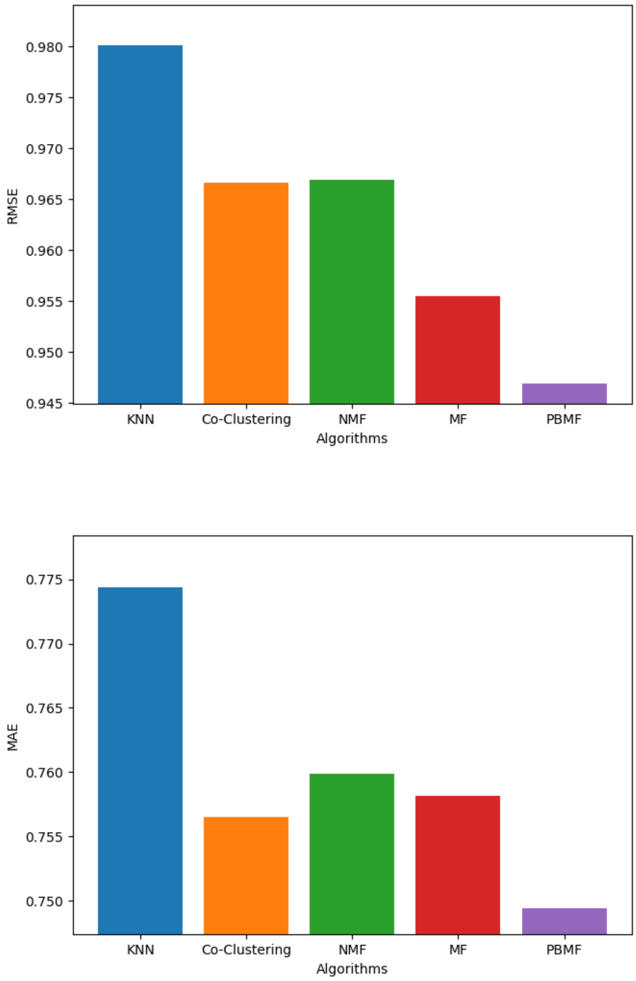

Fig. 1 RMSE comparison of all algorithms

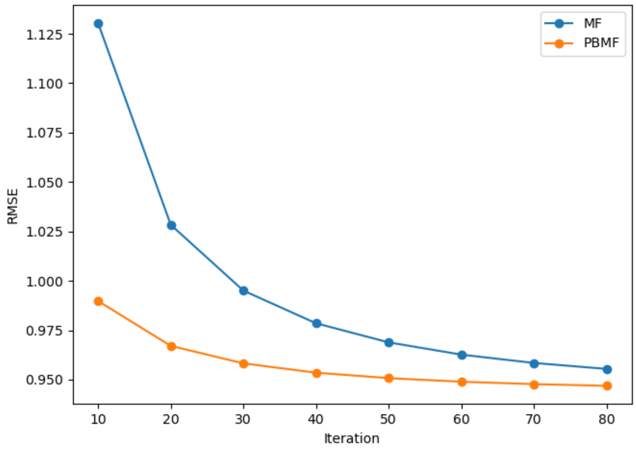

Fig. 2 Comparison of MF and PBMF in different iterations

- Among MF based algorithms, PBMF has the best performance with RMSE of 0.9468 which is 0.0086 less than MF and 0.0201 less than NMF. it clearly demonstrates the positive impact of personality information on the performance of MF.

- As we can see in figure 1 and also in table 2, the difference between RMSE and MAE of the co-clustering algorithm is more than other algorithms. from this larger difference it can be infered that the variance of individual errors in the sample is more than others.
- Figure 2 shows the comparison of RMSE and MAE for MF and PBMF algorithms through different iterations. as we can see the huge difference between two methods start from the basic steps and it gets less as it converges, so it can be said that for any number of iteration PBMF gives better results than MF.

5.3 Hyperparameter tuning

Hyperparameters are parameters of a function which should be specified before starting the learning process and hyperparameter tuning is the process of finding the set of hyperparameters that gives the best result.
In order to get the best result using PBMF we should find the optimum regularization terms and number of latent factors. We use the grid search process for tuning the hyperparameters. To do so we use the whole dataset as trainset and calculate train MSE for each combination of hyperparameters in a 20 iteration learning loop. The results are shown in figure 3.

As we can see, the MSE decreases as the number of factors get higher and then increases by the number of latent factors more than 3. The MSE also decreases as regularization term increases and then increases by the regularization term more than 4. So the optimal

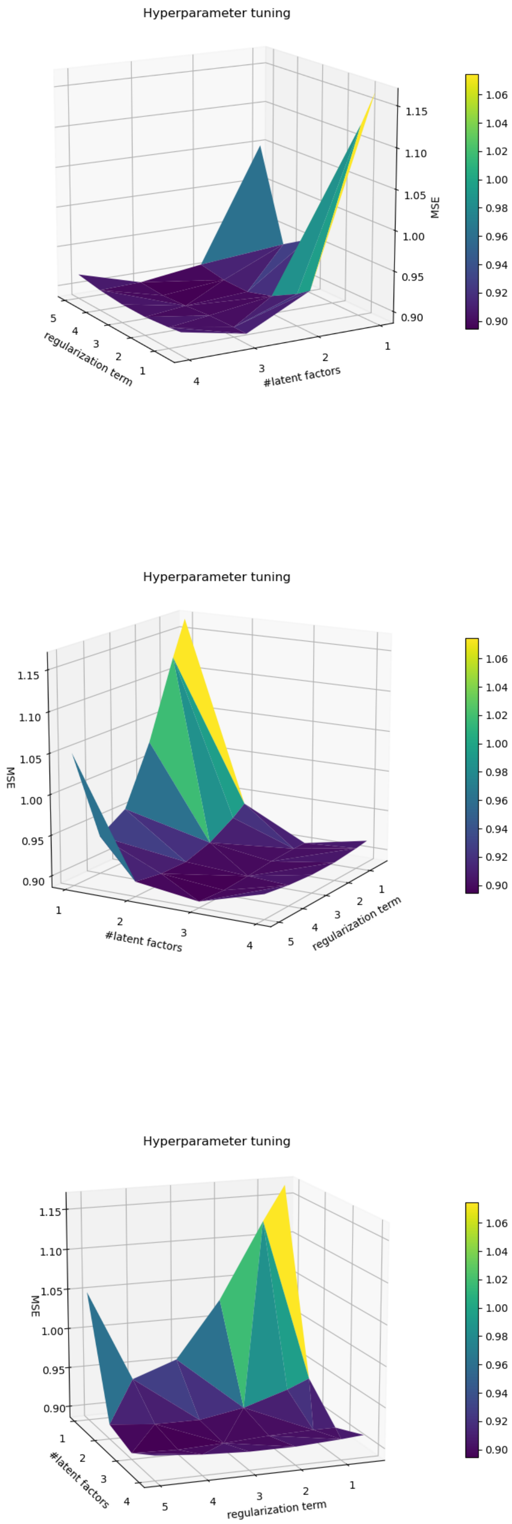

Fig. 3 Hyperparameter tuning

regularization term and number of latent factors that we used in our experiments are 4 and 3 respectively.

6 Conclusion

In this paper we have proposed a new matrix factorization based collaborative filtering algorithm for recommender systems named Personality-Boosted Matrix factorization (PBMF). Experiments on Movielens 100k dataset showed the ability of personality information in terms of optimism and pessimism in empowering the matrix factorization algorithm.
For further studies there are a number of paths considering the power of personality information: 1- investigation of minimum rates that a user should give to be considered in the algorithm could show interesting results. 2- we are also interested in combining personality information in other types of recommender systems.

References

1. Beigi, Ghazaleh, Jiliang Tang, and Huan Liu. "Signed link analysis in social media networks." In 10th International Conference on Web and Social Media, ICWSM 2016, pp. 539-542. AAAI Press (2016)
2. Aggarwal, Charu C. Recommender systems. Vol. 1. Cham: Springer International Publishing, (2016).
3. Schafer, J. Ben, Dan Frankowski, Jon Herlocker, and Shilad Sen. "Collaborative filtering recommender systems." In The adaptive web, pp. 291-324. Springer, Berlin, Heidelberg, (2007).
4. Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5, no. 4 (2015): 1-19.
5. Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Minnesota Univ Minneapolis Dept of Computer Science, (2000).
6. Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42, no. 8 (2009): 30-37.
7. Hu, Rong, and Pearl Pu. "Enhancing collaborative filtering systems with personality information." In Proceedings of the fifth ACM conference on Recommender systems, pp. 197-204. (2011).
8. Koren, Yehuda. "Factor in the neighbors: Scalable and accurate collaborative filtering." ACM Transactions on Knowledge Discovery from Data (TKDD) 4, no. 1 (2010): 1-24.
9. Mehta, Rachana, and Keyur Rana. "A review on matrix factorization techniques in recommender systems." In 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), pp. 269-274. IEEE, (2017).
10. Gao, Rui, Bibo Hao, Shuotian Bai, Lin Li, Ang Li, and Tingshao Zhu. "Improving user profile with personality traits predicted from social media content." In Proceedings of the 7th ACM conference on recommender systems, pp. 355-358. (2013).
11. Cheng, Chen, Fen Xia, Tong Zhang, Irwin King, and Michael R. Lyu. "Gradient boosting factorization machines." In Proceedings of the 8th ACM Conference on Recommender systems, pp. 265-272. (2014).
12. Feng, He, and Xueming Qian. "Recommendation via user's personality and social contextual." In Proceedings of the 22nd ACM international conference on Information Knowledge Management, pp. 1521-1524. (2013).
13. Abdi, Mohamed Hussein, George Okeyo, and Ronald Waweru Mwangi. "Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey." (2018).
14. Luo, Xin, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems." IEEE Transactions on Industrial Informatics 10, no. 2 (2014): 1273-1284.
15. George, Thomas, and Srujana Merugu. "A scalable collaborative filtering framework based on co-clustering." In Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 4-pp. IEEE, (2005).
16. Beigi, Ghazaleh, Suhas Ranganath, and Huan Liu. "Signed link prediction with sparse data: The role of personality information." In Companion Proceedings of The 2019 World Wide Web Conference, pp. 1270-1278. (2019).