Missing Data Modeling with User Activity and Item Popularity in Recommendation

Chong Chen, Min Zhang⋆, Yiqun Liu, and Shaoping Ma

Department of Computer Science and Technology,
Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
z-m@tsinghua.edu.cn

Abstract. User feedback such as movie watching history, ratings and consumptions of products, is valuable for improving the performance of recommender systems. However, only a few interactions between users and items can be observed in implicit data. The missing of a user-item entry is caused by two reasons: the user didn't see the item (in most cases); or the user saw but disliked it. Separating these two cases leads to modeling missing interactions at a finer granularity, which is helpful in understanding users' preferences more accurately. However, the former case has not been well-studied in previous work. Most existing studies resort to assign a uniform weight to the missing data, while such a uniform assumption is invalid in real-world settings. In this paper, we propose a novel approach to weight the missing data based on user activity and item popularity, which is more effective and flexible than the uniform-weight assumption. Experimental results based on 2 real-world datasets (Movielens, Flixster) show that our approach outperforms 3 state-of-the-art models including BPR, WMF, and ExpoMF.

Keywords: Recommender Systems, Collaborative Filtering, Matrix Factorization, Implicit Feedback

1 Introduction

In the era of information explosion, not only are users not easy to find items they are interested in, such as news, merchandise, music, etc., it is also difficult for providers to display products accurately to the target population. In this case, the role of recommender systems is becoming increasingly important.

The key to recommender systems is to infer users' preferences from historical records, such as ratings, reviews, clicks, and consumptions, etc. Compared to explicit feedback (e.g., ratings and reviews), implicit feedback like users' video viewing and product purchase history, doesn't require users' extra operations and can be tracked automatically. Therefore it is much easier for providers to

⋆Corresponding author

collect. However, implicit feedback is more challenging to utilize, since it is binary and only has positive examples. When inferring users' preferences, the items without interactions are essential. These items are referred to as missing data in recommender systems.

Previous studies [6,7,12,15]deal with this problem in two ways: either randomly sampling negative instances from the missing data, or treating all of them as negative. However, an important fact cannot be overlooked when we revisit this problem — many items, actually a large number of them were not clicked because the user didn't see them, rather than disliked them. If an item was never noticed by the user, then no consumption can possibly be made, and the missing interaction implies no particular positive or negative preference of the user at all. Many previous approaches have not distinguished these two cases. They assign a uniform weight to the missing data, assuming that the missing entries have equal probability to be negative feedback, and hence we prefer that these studies are biased in terms of modeling users' preferences accurately.

Compared with the previous work, [6,10]propose to weight the missing data based on item popularity. In the same condition, popular items are more likely to be known by users in general [4], and thus it is reasonable to think that a missing popular item is more probable to be truly not attracted (as opposed to unknown) to the user. Using item popularity to model missing data is effective, but it still has a flaw: not considering the differences among different users and making all users have the same weight to a missing item.

In fact, users with different degrees of activity usually have different visibility for items. Inactive users (or new users) tend to browse popular items, while active users are more likely to browse unpopular items. In this paper, we define user activity as the total number of items clicked by the user, while item popularity is defined as the number of users who clicked on it. In Fig. 1, we show the relationship between user activity and item popularity on Flixster dataset. As we can see from the figure, the curve shows an obvious downward trend, which indicates that unpopular items are more likely to be known by active users.

In this paper, we propose a new method named UIMF, considering both user activity and item popularity to weight missing data and make recommendation for implicit feedback. Experiments have been conducted on 2 real-world datasets in comparison with 3 state-of-art approaches. The encouraging results verify the effectiveness of our model.

The remainder of the paper is organized as follows. The next section introduces relevant prior work on collaborative filtering and implicit feedback. Section 3 gives a detailed introduction about modeling missing data based on user activity and item popularity. We conduct experiments and the results are presented in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

User feedback is frequently seen in real-life scenarios and is usually in different forms, such as ratings, reviews, clicks and consumptions of items. Handling user

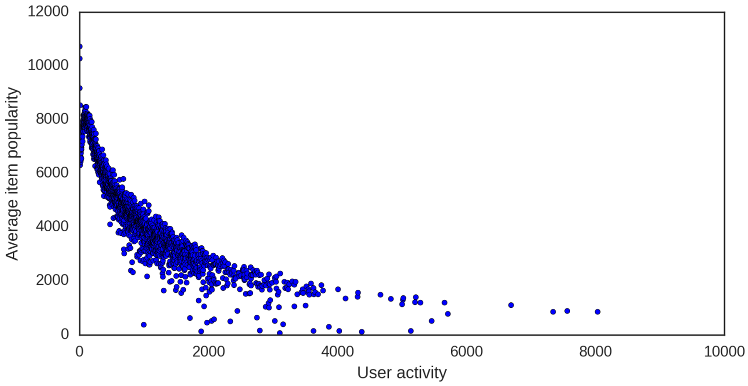

Fig. 1. The relationship between user activity and item popularity on Flixster dataset

feedback has been a key issue in recommender systems. Many studies have been made to enhance the performance of recommendation through these historical records.

In recent years, matrix factorization (MF) has become the most popular collaborative filtering approach [9,16]. The original MF models were designed to model users' explicit feedback by mapping users and items to a latent factor space, such that user-item relationships (e.g., ratings) can be captured by their latent factors' dot product. Based on that, many research efforts have been devoted to enhancing MF, such as integrating it with neighbor-based models [8] and extending it to factorization machines [14] for a generic modeling of features. However, it is still problematic to apply traditional matrix factorization to implicit feedback due to the lack of negative instances.

To this end, two basic strategies have been proposed in previous studies [6]: sample based learning that samples negative instances from the missing data [12, 15] and whole-data based learning that sees all the missing data as negative [7, 11,18]. Compared with sample based methods, whole-data based methods can model the full data with a potentially higher coverage, and thus may achieve a better performance if the parameters are set properly.

Most existing whole-data based methods [3,7,11,13,17,18] assign a uniform weight to all the missing data, assuming that the missing entries have the same probability to be negative feedback, which facilitates the efficiency of the algorithm, but limits the flexibility and extensibility of the model. As discussed in Section 1, [6,10] are the only works that consider item popularity for weighting missing feedback. [10]proposes the concept of exposure to model whether an item has been observed by a user. It thinks that popular items are more likely to be known by users and thus gives a higher weight to popular items. [6]devises a new object function of matrix factorize, in which item popularity is used to model the confidence that item i missed by users is a truly negative instance. Unlike the previous two methods, [15] adopts popularity-based oversampling for

Table 1. Variables introduction

Variables	Meaning
y_{ui}	User-Item Interaction: i.e. whether user u has clicked on item i
θ_u	The preference latent factor vector of user u
β_i	The attribute latent factor vector of item i
a_{ui}	Whether user u has seen item i
p_{ui}	The probability that user u sees item i
μ_i	The parameter of "item popularity only" strategy
η_u	The parameter of "user activity only" strategy
ω_{ui}	The parameter of "both user activity and item popularity" strategy
(α_1, α_2)	The parameter of Beta distribution

learning BPR, which basically samples popular items as negative feedback with a higher probability.

In our work, we propose a novel approach to model missing data and utilize Bayesian approaches to estimate the weight of them. Differing from previous studies, our work uses both user activity and item popularity in modeling missing data, while only item popularity is considered in [6,10,15].

To our knowledge, this work is the first attempt to exploit user activity for modeling missing data.

3 Missing Data Modeling

In this section, we present our model (UIMF). In Section 3.1, we briefly introduce the model of modeling missing data with user activity and item popularity. In Section 3.2, we derive inference procedures for our model. The connections between our approach and other models are shown in Section 3.3. The variables of our model are listed in Table. 1

3.1 Model Description

As implicit data is very sparse, the interactions between users and items that can be observed are rather limited. The variable y_{ui} indicates whether there is an interaction between user u and item i (if there is an interaction, $y_{ui} = 1$, otherwise $y_{ui} = 0$). The general idea of this model is that, many items were not clicked or consumed not because the user didn't like them, but because the user didn't see them. When inferring users' preferences, we need to assign an appropriate weight to each missing entry according to the probability that the user sees the item. We use a_{ui} to indicate whether user u has seen item i ($a_{ui} = 1$ means that u has seen i, and otherwise $a_{ui} = 0$). Then the variable p_{ui} is introduced to capture the probability that $a_{ui} = 1$. If p_{ui} is large, very possible that user u has seen item i but choose not to click on it, then the confidence that i is a truly negative instance should also be large (The converse argument also holds for low values of p_{ui}).

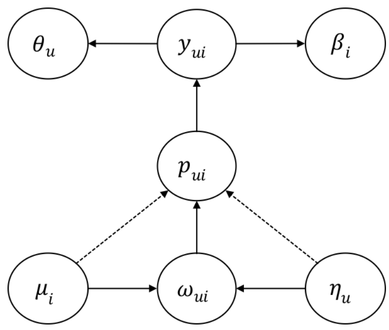

Fig. 2. Graphical representation of our MF model. A directed edge from node a to node b denotes that the variable b depends on the value of variable a. μ_i and η_u are derived from item popularity and user activity respectively. ω_{ui} is derived from μ_i and η_u and uses both user activity and item popularity.

The value of p_{ui} is related to the popularity of item i and the activity of user u. If an item is popular, then it is more likely to be seen by users. Similarly, if a user is active, then the probabilities that he sees items are higher. Therefore, the weight of the missing entry $(y_{ui}, y_{ui} = 0)$ should be assigned large if u is active and i is popular.

Based on the above idea, we propose a new method for matrix factorization. The graphical model is presented in Fig. 2. Given the condition that user u has seen item i $(a_{ui} = 1)$, the probability that user u would click on item i follows Gaussian distribution [16] (λ_y is precision for corresponding Gaussian distribution):

$$y_{ui}|(a_{ui} = 1) \sim N\left(\theta_u^T \beta_i, \lambda_y^{-1}\right), \forall u, i \tag{1}$$

and the variable a_{ui} follows Bernoulli distribution:

$$a_{ui} \sim Bernoulli(X), \forall u, i \tag{2}$$

where X can be replaced by μ_i, η_u and ω_{ui}, which represent the priors of the Bernoulli distribution derived from "item popularity only", "user activity only" and "both user activity and item popularity".

In Fig. 3, we show an example of using different strategies to weight missing data. The strategy of "item popularity only" makes every user has the same weight to a missing item, while "user activity only" makes every missing item has no difference for a user. These two methods do not correspond with the actual situation. Different from them, in our model we propose to weight every missing entry individually by considering both user activity and item popularity, which is more practical in real-word settings.

However, one thing can not be ignored : the high space complexity makes it impossible to explicitly store the weight matrix W even for medium-sized datasets. As an alternative, we firstly capture μ_i and η_u from Beta distributions

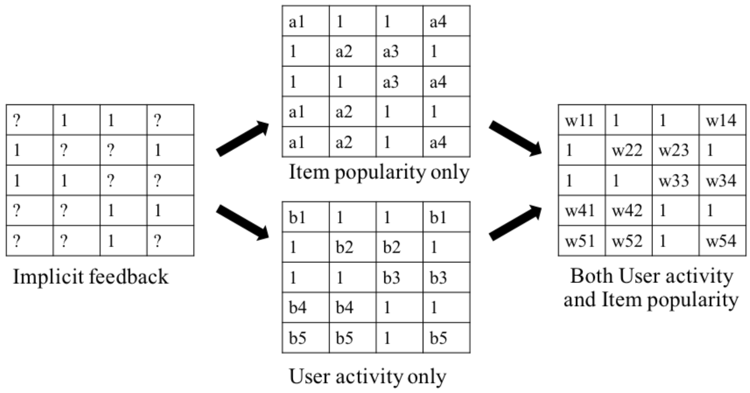

Fig. 3. Different strategies to weight missing data. a_i,b_u and w_{ui} denote the weight of missing entry and we use $W = [w_{ui}]_{M\times N}$ to represent the weight matrix.

respectively:

$$\mu_i \sim Beta(\alpha_1, \alpha_2), \forall i; \eta_u \sim Beta(\alpha_1', \alpha_2'), \forall u \tag{3}$$

After that, we construct the necessary part of ω_{ui} when it is used to update the user/item factors by adding these two variables with individual weights.

Our proposed model is named UIMF, as it considers both user activity and item popularity for modeling missing feedback.

3.2 Model Learning

We use expectation-maximization (EM) [2] algorithm to infer the parameters of UIMF.

The E step that can be derived from Bayesian Theorem is:

$$p_{ui}\left(y_{ui}=0\right) = \frac{\omega_{ui}\cdot N\left(0|\theta_u^T\beta_i, \lambda_y^{-1}\right)}{\omega_{ui}\cdot N\left(0|\theta_u^T\beta_i, \lambda_y^{-1}\right) + (1-\omega_{ui})} \tag{4}$$

where $N\left(0|\theta_u^T\beta_i, \lambda_y^{-1}\right)$ stands for the probability density function of $N\left(\theta_u^T\beta_i, \lambda_y^{-1}\right)$ evaluated at 0. Since p_{ui} indicates the probability that user u sees item i, we can define $p_{ui}(y_{ui}=1)=1$.

Therefore, we can estimate the user/item factors in the following M step, which can be derived based on Alternative Least Square (ALS) optimization (λ_s is corresponding precision matrix for Gaussian distribution; I_k is the identity matrix.):

$$\theta_u \leftarrow \left(\lambda_y \sum_i p_{ui}\beta_i\beta_i^T + \lambda_\theta I_k\right)^{-1}\left(\sum_i \lambda_y p_{ui} y_{ui}\beta_i\right) \tag{5}$$

$$\beta_i \leftarrow \left(\lambda_y \sum_u p_{ui}\theta_u\theta_u^T + \lambda_\theta I_k\right)^{-1} \left(\sum_u \lambda_y p_{ui} y_{ui} \theta_u\right) \tag{6}$$

We update μ_i, η_u and ω_{ui} as follows:

– **Item popularity only:**

$$\mu_i \leftarrow \frac{\alpha_1 + \sum_u p_{ui} - 1}{\alpha_1 + \alpha_2 + \|U\| - 2} \tag{7}$$

– **User activity only:**

$$\eta_u \leftarrow \frac{\alpha_1{}' + \sum_i p_{ui} - 1}{\alpha_1{}' + \alpha_2{}' + \|I\| - 2} \tag{8}$$

– **Both user activity and item popularity:**

$$\omega_{ui} \leftarrow k \cdot \mu_i + (1-k) \cdot \eta_u \tag{9}$$

To make recommendation, we consider both the probability that the user sees items and his preference (an item is more likely to be seen means more likely to be clicked). Therefore, we construct the following ranking score:

$$\bar{y_{ui}} = p_{ui} * \theta_u^T \beta_i \sim \omega_{ui} * \theta_u^T \beta_i \tag{10}$$

and the items (unclicked/ uncomsumed) are ranked in descending order $\bar{y_{ui}}$ to provide the Top-N item recommendation list.

3.3 Model Flexibility.

Our proposed model can be easily converted to other models by changing the update method of ω_{ui}.

When the value of ω_{ui} is fixed to 1, we recover traditional matrix factorization [16]. When it is fixed between 0 and 1, we can get weighted matrix factorization (WMF) [7]. ExpoMF [10] is also a special case of our model which can be obtained by fixing the value of k to 1 in Equation (9) when updating ω_{ui} and using "item popularity only" strategy.

4 Experiment

We begin by introducing the experimental settings. Then we present the experimental results conducted on 2 real-world datasets, followed by an exploratory analysis on the influence of user activity and item popularity.

Table 2. Statistics of the evaluation datasets.

Datasets	#users	#items	#interactions	#density
MovieLens	6,040	3,706	1,000,209	4.47%
Flixster	147,229	17,318	8,093,735	0.317%

4.1 Experimental Settings

Datasets. We evaluate on 2 real-world datasets: MovieLens[1] and Flixster[2]. Movielens has been widely used to evaluate collaborative filtering algorithms, the version we used contains about one million ratings. Flixster is a dataset for evaluating social information based recommendation, it has a huge amount of interactions from around one hundred thousand users and ten thousand items. The datasets have been preprocessed so that all the items have at least 5 ratings. We treat the corresponding rating as 1 as long as there is a user-item interaction, which is the same procedure adopted in many previous studies including [5–7,10,15]. The statistical details of the 2 datasets are presented in Table 2.

Baselines and our Methods. We compare with the following recommendation methods for implicit feedback:

- **BPR** [15]: This is a sample based method that optimizes the pair-wise ranking between the positive and negative samples.
- **WMF** [7]: This is a whole-data based method that treats all missing interactions as negative instances and weights them uniformly.
- **ExpoMF** [10]: This is a state-of-the-art whole-data based method for item recommendation. It also treats all missing interactions as negative instances but weights them non-uniformly by item popularity.

We also compare the performance of 3 missing data modeling strategies in our model. Because the strategy of “item popularity only” is the same as ExpoMF, we present the following 2 methods:

- **UIMF**:This is the model we proposed in this paper, which takes both user activity and item popularity into consideration to weight missing interactions.
- **UMF**: This is a special case of our model, which uses the strategy of “user activity only” to weight missing interactions.

Evaluation Metrics. We adopt frequently used metrics [1] to evaluate the performance, including Recall@K, MAP@K and NDCG@K. where $rel_i = 1/0$ indicates whether the item at rank i in the Top-N list is in the testing set. For

[1] http://grouplens.org/datasets/movielens/1m/
[2] http://www.sfu.ca/~sja25/datasets/

Table 3. Performance comparison on 2 datasets. The best performing method is boldfaced, and the last column shows the improvements of UIMF compared to the best results in baselines and UMF. The improvements with "*" are significant with p-value < 0.05, and the improvements with "**" are significant with p-value < 0.01

MovieLens	**BPR**	**WMF**	**ExpoMF**	**UMF**	**UIMF**	***UIMF vs. best***
Recall@10	0.1442	0.4241	0.4250	0.4259	**0.4267**	0.18%
Recall@50	0.3876	0.4704	0.4734	0.4720	**0.4753****	0.40%
Recall@100	0.5340	0.5686	0.5723	0.5700	**0.5743****	0.34%
NDCG@10	0.4190	0.4473	0.4481	0.4495	**0.4502**	0.15%
NDCG@50	0.4071	0.4343	0.4363	0.4361	**0.4383****	0.47%
NDCG@100	0.4462	0.4721	0.4745	0.4739	**0.4766****	0.44%
MAP@10	0.2864	0.3076	0.3087	0.3100	**0.3102**	0.06%
MAP@50	0.2077	0.2290	0.2305	0.2307	**0.2320****	0.64%
MAP@100	0.2082	0.2310	0.2327	0.2327	**0.2344****	0.72%
Flixster	**BPR**	**WMF**	**ExpoMF**	**UMF**	**UIMF**	***UIMF vs. best***
Recall@10	0.1622	0.3766	0.3788	0.3826	**0.3865***	1.02%
Recall@50	0.3337	0.4909	0.4947	0.4975	**0.5029****	1.09%
Recall@100	0.4366	0.5596	0.5684	0.5659	**0.5727****	1.20%
NDCG@10	0.1227	0.3266	0.3248	0.3302	**0.3323***	0.64%
NDCG@50	0.1765	0.3534	0.3543	0.3594	**0.3638****	1.22%
NDCG@100	0.2061	0.3725	0.3760	0.3792	**0.3845****	1.40%
MAP@10	0.0893	0.2489	0.2484	0.2511	**0.2529***	0.72%
MAP@50	0.1005	0.2355	0.2379	0.2408	**0.2453****	1.87%
MAP@100	0.1046	0.2351	0.2391	0.2441	**0.2466****	1.02%

each user, these metrics can be computed as follows (each metric is the average for all users, and MAP is the average of all AP of users).

$$\begin{aligned} Recall@K &= \frac{\sum_{i=1}^{K} rel_i}{min\left(K, y_u^{test}\right)}; AP@K = \sum_{n=1}^{K} \frac{\frac{\sum_{i=1}^{n}}{n} \times rel_n}{min\left(K, y_u^{test}\right)} \\ DCG@K &= \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{log_2\left(i+1\right)}; NDCG@K = \frac{DCG@K}{IDCG@K} \end{aligned} \tag{11}$$

To evaluate on different recommendation lengths, we set K = 10, 50 and 100 in our experiments.

Experiments Details. We adopt the open source implementation in librec[3] to obtain the predictions of BPR; for WMF we use the open source code from github[4] since it can get a better performance than librec; and for ExpoMF, we use the source code [5] released by the authors. The parameters for baseline methods

[3] https://www.librec.net
[4] https://github.com/benanne/wmf
[5] https://github.com/dawenl/expo-mf

are initialized as in the corresponding paper and they are further carefully tuned around to achieve the best performance. The dimensions of latent factor vectors are set to 50 for both MovieLens and Flixster.

4.2 Performance Comparison

We perform a four-fold cross-validation in our experiments. Three folds are used for training and the rest fold is used for testing. For every dataset, we conduct all methods eight times and the average result is presented in Table 3.

We make the following observations:

First, as shown in Table 3, The methods weighting missing data non-uniformly (ExpoMF, UMF, UIMF) generally outperform the uniform weighting method WMF. We believe the benefits mainly come from the non-uniform setting of weight that derived from user activity or item popularity since it is more practical in real-world.

Secondly, our method (UIMF) using both user activity and item popularity outperforms ExpoMF and UMF, which weight missing data only by item popularity or user activity. This is because that UIMF weights each missing entry individually so that it can better capture users' preferences and items' attributes as described in Section 3.1. We also observe that UIMF tends to show more obvious improvement on Flixster than MovieLens. We think the reason is that Flixster is sparser and thus has more missing entries than MovieLens, which can be utilized better by UIMF since it is designed for dealing with the problem of missing data.

Another observation is that UIMF shows no significant improvement ($p<0.05$) on MovieLens dataset when the value of N is set to 10 in Top-N recommendation. However, as the value of N is set higher, the improvement of UIMF becomes more obvious, showing that our model is more accurate when it comes to covering a wider range of users' interests.

4.3 Impact of User Activity and Item Popularity

The value of the variable k in Equation 9 determines the weights of user activity and item popularity when modeling missing data. To explore the impact of these two parts, we conduct experiments on different values of k in Equation 9 when updating ω_{ui}. We alter the value of k with a stepsize of 0.1, and compare the performances correspondingly. Note that a value of 1 corresponds to ExpoMF and a value of 0 corresponds to UMF. The results of Recall@50, NDCG@50 and MAP@50 on MovieLens dataset are presented in Fig. 4.

From the figure above, we can first see that UIMF generally outperforms the baseline method WMF. Secondly, compared with the strategies of "user activity only" ($k = 0$) and "item popularity only" ($k = 1$), using both of them (no matter how much the value of k is adopted from 0 to 1) can always get a better performance, indicating the effectiveness of using both user activity and item popularity for missing data modeling.

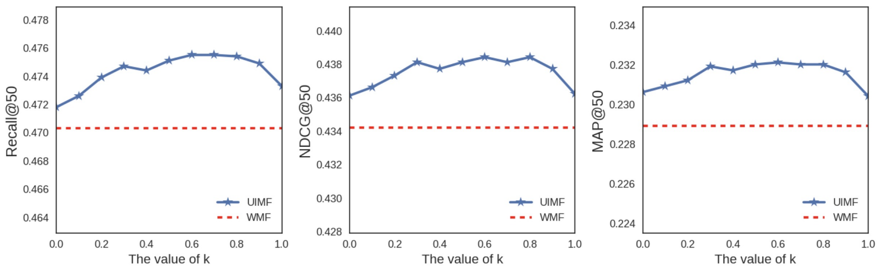

Fig. 4. The Recommendation Performances with different values of k on MovieLens dataset.

What's more, the performances increase with the increase of variable k from 0 to 0.6, and then reach the optimal performance at around 0.6, after that, the performances gradually decline with the value of k increases from 0.8 to 1. This illustrates that user activity and item popularity may not be equally impactful when weighting missing data, and a proper value of k is needed to better combine them so that UIMF can achieve the best performance.

5 Conclusion

In this paper, we study the problem of how to model missing data in recommendation. Different from previous work that applied a uniform weight on missing interactions or just weighted them based on item popularity, we propose to consider both user activity and item popularity to weight missing data. A novel unified model(UIMF) is designed based on this idea. The major contributions of this work are:

First, we propose to consider both user activity and item popularity to model the missing data, which helps to capture users' preferences and items' attributes more accurately. As far as we know, this work is the first attempt to exploit the impact of user activity for implicit data in the literature.

Second, we design a novel unified model(UIMF) to connect both user activity and item popularity and use the variable k to control the influence of each part.

Third, extensive experiments have been conducted on 2 real-world datasets in comparison with 3 previous methods. Statistically our proposed model (UIMF) achieves significantly better performances in most cases, which verifies the effectiveness of the model.

In the future, we will make further improvements to the model to address the problem of high time complexity due to the non-uniform weighting of missing data.

6 Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the Natural Science Foundation of China under Grant No.: 61672311 and 61532011.

References

1. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 39–46. ACM (2010)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) pp. 1–38 (1977)
3. Devooght, R., Kourtellis, N., Mantrach, A.: Dynamic matrix factorization with priors on unknown values. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 189–198. ACM (2015)
4. He, X., Gao, M., Kan, M.Y., Liu, Y., Sugiyama, K.: Predicting the popularity of web 2.0 items based on user comments. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. pp. 233–242. ACM (2014)
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
6. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 549–558. ACM (2016)
7. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. pp. 263–272. Ieee (2008)
8. Koren, Y.: 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. ACM (2008)
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8) (2009)
10. Liang, D., Charlin, L., McInerney, J., Blei, D.M.: Modeling user exposure in recommendation. In: Proceedings of the 25th International Conference on World Wide Web. pp. 951–961. International World Wide Web Conferences Steering Committee (2016)
11. Ning, X., Karypis, G.: Slim: Sparse linear methods for top-n recommender systems. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on. pp. 497–506. IEEE (2011)
12. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. pp. 502–511. IEEE (2008)
13. Pilászy, I., Zibriczky, D., Tikk, D.: Fast als-based matrix factorization for explicit and implicit feedback datasets. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 71–78. ACM (2010)

14. Rendle, S.: Factorization machines. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. pp. 995–1000. IEEE (2010)
15. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 452–461. AUAI Press (2009)
16. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Nips. vol. 1, pp. 2–1 (2007)
17. Steck, H.: Training and testing of recommender systems on data missing not at random. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 713–722. ACM (2010)
18. Volkovs, M., Yu, G.W.: Effective latent models for binary feedback in recommender systems. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 313–322. ACM (2015)