Expert Systems With Applications 213 (2023) 118932

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Selection bias mitigation in recommender system using uninteresting items based on temporal visibility

Lei Shi [a], Shuqing Li [a,*], Xiaowei Ding [b,*], Zhan Bu [a]

[a] College of Information Engineering, Nanjing University of Finance & Economics, Nanjing 210023, China
[b] School of Information Management, Nanjing University, Nanjing 210023, China

ARTICLE INFO

Keywords:
Collaborative filtering
Explicit feedback
Selection bias
Pre-use preference
Uninteresting items

ABSTRACT

Most collaborative filtering recommendation algorithms rely too much on the user's historical rating data. However, selection bias is common in explicit feedback data, which makes the learning of user preferences face more challenges. We verify the influence of selection bias on topN recommendation, and propose a data filling strategy using uninteresting items based on temporal visibility to alleviate the selection bias in the data. Specifically, our method includes a weighted matrix factorization model to learn users' pre-use preferences for unrated items. According to the experience of items that users have seen but not interacted show negative preferences, we combine user activity, item popularity and temporal rating information to carry out non-uniform weighting to evaluate the confidence of unrated items as a negative example. Then the items with low pre-use preferences are taken as uninteresting items and filled in a low value to restore the user's real rating distribution. Experiments on two real world datasets show that our algorithm can effectively alleviate the selection bias and improve the recommendation accuracy.

1. Introduction

At the times of information explosion, recommender system plays an important role in alleviating information overload. It has been widely used by many online services such as e-commerce, online news and social media websites. It is also a widely concerned research in academia and industry (Lu et al., 2015).Collaborative filtering is one of the most widely used and deeply studied algorithms in recommender system. Collaborative filtering algorithm learns user preferences and makes recommendations based on user historical behavior data (Ha & Lee, 2017). User behavior data can be divided into two categories: explicit feedback data (such as rating, comments) and implicit feedback data (such as purchase, click). Explicit feedback data is mainly used for rating prediction, most algorithms try to directly model the observed rating data to predict the unrated items. However, the selection bias of explicit feedback data makes the recommendation algorithm trained based on such data unreliable on topN recommendation. For example, user will only rate a movie if he/she has been exposed to the movie. Due to the large number of movies and the wide use of recommendation algorithms, each user may not have equal access to each movie, which will lead to the missing value in the rating matrix not at random (Marlin & Zemel, 2009). This is referred to as the Missing-Not-At-Random (MNAR) problem (Marlin et al., 2007; Marlin & Zemel, 2009). In short, the method based on explicit feedback cannot be directly used in topN recommendation scenarios in which selection bias must be effectively corrected.

Recently, most algorithms use implicit feedback data for topN recommendation.Implicit feedback data considers both observed data and missing data, so it is not affected by selection bias. The advantage of implicit feedback is that it makes full use of the negative preferences implied by missing data (Jawaheer, Szomszor & Kostkova, 2010). However, compared with explicit feedback, implicit feedback is not clear about the expression of user preferences and cannot express the degree of preferences (Xue et al., 2017). In other words, the method based on implicit feedback can not guarantee that users have high post-use preference for recommended items (Sarkar, Mitsui, Liu, & Shah, 2020). Whether the pre-use preference expressed by implicit feedback can be applied to explicit feedback to alleviate selection bias is worth discussing.

Therefore, in this work we address the following research questions:

RQ1: How does the selection bias of explicit feedback affect the data distribution and recommendation results?

* Corresponding authors.
E-mail addresses: leeshuqing@nufe.edu.cn (S. Li), dingxiaowei@nju.edu.cn (X. Ding), buzhan@nuaa.edu.cn (Z. Bu).

https://doi.org/10.1016/j.eswa.2022.118932
Received 19 November 2021; Received in revised form 18 September 2022; Accepted 25 September 2022
Available online 30 September 2022
0957-4174/© 2022 Elsevier Ltd. All rights reserved.

RQ2: Whether modeling pre-use preferences with implicit feedback information and temporal rating information can effectively identify uninteresting items?

RQ3: Whether the filling strategy based on uninteresting items can alleviate the selection bias? Whether the explicit feedback method of alleviating the selection bias is more effective than the implicit feedback method?

In this paper, we demonstrate that selection bias will lead to the bias of rating distribution towards high rating, and its impact on recommendation accuracy is obvious. We propose a weighted matrix factorization model based on implicit feedback to identify uninteresting items, combine user activity, item popularity and temporal rating information to carry out non-uniform weighting. Our experimental show that our filling strategy can effectively alleviate the influence of selection bias, significantly improve the accuracy of topN recommendation, and its performance is better than the method using only implicit feedback.

The goal of this work is to provide a general data filling strategy for the method based on explicit feedback to alleviate the selection bias and make it suitable for topN recommendation. The remainder of this paper is structured as follows. Section 2 analyses the limitations of existing collaborative filtering algorithms in topN recommendation and existing related works. Section 3 introduces our algorithm in detail. Section 4 carries out experiments and corresponding analysis, and section 5 summarizes the results and provides an outlook on further work.

2. Related work

2.1. Limitation of explicit feedback method

The collaborative filtering algorithm based on explicit feedback has obvious differences in rating prediction and topN recommendation. The algorithm that works well in the rating prediction is not necessarily effective in the topN recommendation. Steck (2013) found that the main difference between the two scenarios lies in the training and test data. Rating prediction is only related to the observed ratings. In order to evaluate the accuracy of the rating prediction, the existing studies often predict only rated items to calculate the error. TopN recommendation often need to predict and rank all missing ratings. Learning user preferences only from the observed ratings can accurately predict the actual rated items, but it is not enough to effectively predict all unrated items. The main reason lies in the sparsity of ratings and user selection bias.

Firstly, data sparsity is a common challenge for collaborative filtering algorithms. In the real world, users often rate only a small number of items, and most items lack ratings (Da Silva, De Moura Junior & Caloba, 2018). Collaborative filtering algorithm is difficult to accurately learn users' real preferences. Secondly, the observed ratings has user selection bias, and users tend to choose items that may bring them high satisfaction and ignore those items that may bring low satisfaction (Chen et al., 2020). In other words, the result of users' choice and the observed ratings are missing not at random (Chen, Yeh, & Ma, 2021). This leads to high rating in most of the observed ratings and low ratings in only a small part. The rating distribution of the data set used in this paper also conforms to this situation. In contrast, many researches have shown that in the real world, users are often only interested in a small number of items, and most items are not interested (Hwang et al., 2016). Due to user selection bias, observed ratings are not a representative sample of all ratings, which tend to reflect users' positive preferences. Many items that may indicate users' negative preferences are not utilized due to lack of rating.

The existing collaborative filtering algorithms often only consider the observed ratings, ignore the impact of data sparsity and user selection bias. It is difficult to balance the positive and negative preferences of users, and there is bias in the prediction rating and ranking of missing data, resulting in poor topN recommendation effect. Therefore, how to effectively mine and utilize the user preferences implied by missing data is the key of this paper.

2.2. Existing work analysis

For the use of missing data, data filling is a direct and effective means. The simplest filling method is to fill the rating matrix with the rating mean and mode of users or items (Cheng, Feng & Gui, 2019), but the filling value is single and the difference between users and items is ignored, so the reliability is not strong. Many researches implement rating prediction algorithms based on observed ratings, and the missing items are filled in with the predicted value of the rating prediction algorithm. Ma, King & Lyu (2007) proposed an effective missing data prediction method (EMDP), which fills in missing values in combination with the prediction rating based on user and item based collaborative filtering algorithm, and gives priority to filling in highly reliable data. Ren et al. (2012)proposed automatic filling and adaptive maximum filling methods, which can adaptively consider the neighborhood information from the perspective of users and items to identify and fill in the key missing data in each prediction. Preliminary Data-based Matrix Factorization (PDMF) proposed by Yuan et al. (2021) generates preliminary prediction data based on neighborhood-based methods, which can make full use of the imputed data to alleviate data sparsity. Although the above methods can effectively alleviate the data sparsity and have good effect in the rating prediction, the effect in the topN recommendation is not ideal, because they all overestimate the rating of missing data.

In recent years, with the further development of data enhancement technology, the traditional data filling methods are also being improved. In fact, some traditional data filling methods such as BPR(Bayesian Personalized Ranking), which are widely used in recommender systems, can be also seen as implicit data enhancement methods. For example, BPR can expand the user vector into user matrix, so as to convert the user item pair into a triplet of the user and two items, thereby indicating the user's preference difference between items (Rendle et al., 2012). In many other recommender system methods, data enhancement methods are also widely combined to produce new effects (Nguyen et al, 2022). For example, CDAE uses autoencoder to increase the potential factor of each user in the input of collaborative filtering, thus enhancing the ability of denoising autoencoder (Wu et al, 2016).

Since the user rating used by the recommender system is a kind of data with various bias, it cannot completely and objectively reflect the real preference of users. Therefore, in recent years, with the continuous deepening of data bias research, exploring effective methods for data enhancement from the perspective of data bias elimination has become a widely concerned research point (Liao et al, 2021). For example, Ashokan discusses different types of bias and proposes various definitions of fairness, and designs a new bias mitigation strategy to solve the potential unfairness of the rating in recommender system (Ashokan & Haas, 2021). Misztal-Radecka and Indurkhya (2021) proves that the generated recommendations are often optimized for the mainstream trend, and proposes a bias aware hierarchical clustering algorithm to detect and describe the user groups that may be discriminated in a given recommendation algorithm. Deldjoo, Bellogin and Di Noia (2021) investigated the impact of recent data characteristics on the performance of classical recommender systems, and proved that it is more difficult to explain the change of performance than accuracy when dealing with the level of fairness. However, little attention has been paid to the selection bias in explicit feedback and its impact on topN recommendation, which is the focus of this paper.

Since most of the existing filling algorithms train the model based on the existing rating data, they often do not take into account the user selection bias of the observed ratings, so most filling ratings are high. Chae et al. (2019) also noticed this problem and found that if the model learns such a high rating from the observed ratings, it will also generate high ratings for most unrated items. Although most unrated items indicate user's negative preference, many existing methods are still likely to fill them with high ratings. These overestimated ratings will further aggravate the phenomenon of high observed ratings. Although this kind of filling algorithm alleviates the data sparsity, it does not take into account the impact of user selection bias, which makes it difficult for the collaborative filtering algorithm to learn the user's real preference for missing data, and the topN recommendation effect is even worse. We will verify this in experiment 4.2.

Of course, some researches recognize that most unrated items indicate negative preferences of users. For example, pureSVD simply uses 0 to fill all unrated items, and then directly performs traditional singular value decomposition (Cremonesi, Koren & Turrin, 2010). Steck (2010) gives all unrated items a uniform low value, and believes that the unrated items carry the user's negative preference information. These methods fill all unrated items with a unified low value, and the topN recommendation effect is significantly improved, because they make full use of the implied negative preference of missing data and substantially alleviate the impact of user selection bias. However, these methods often simply regard all unrated items as uninteresting items, and does not take into account that the unrated items also include items that may be of interest to users. Pan Pan et al. (2021) proposes a deep learning component to learn the dynamic score history embedding of each user to estimate the probability distribution of items scored by users in order. These estimated dynamic exposure probabilities are then used as propensity scores. However, this method relies heavily on the availability of propensity scores and cannot be extended to other algorithms.

3. Principle and design of method

3.1. Basic idea of method

In view of the above analysis, we proposes a data filling method based on uninterested items. Firstly, it accurately identify items in the missing data that are not of interest to the user, and then fill them with low values. It alleviates the data sparsity and solves the influence of user selection bias, so as to effectively enhance the topN recommendation performance. Secondly, the key is how to identify the items in the missing data that the user is not interested in. Unrated items can be caused by the following two reasons (Chen et al., 2021):

1) The user has not seen the item and does not know its existence so that there is no rating.
2) The user saw the item and knew its existence but was not interested in it so that there was no rating.

The second type obviously shows the user's negative preference (Liang et al., 2016). Those items can be defined as uninteresting items. In order to identify the uninteresting items in the missing data, we uses the expression method of pre-use preference to divide users' preference for items into pre-use preference and post-use preference (Hwang et al., 2016). Pre-use preference is the user's impression of the item before use, which determines whether the user interacts with the item. The post-use preference determines the user's rating of the item after actual use.

Elahi Elahi et al. (2021)find that a few popular users/items get more popular and many unpopular users/items get more unpopular. Active users have higher visibility of popular items during their active period. For users, the items with higher visibility are more likely to be uninteresting if they are not interacted. In other words, users have low pre-use preference for those items. Therefore, we proposes a weighted matrix factorization algorithm based on item temporal visibility(Item Temporal Visibility- element-wise Alternating Least Squares, ITV-eALS) to model and train the pre-use preference of missing data. We combine user activity, item popularity and time factors to comprehensively measure the visibility of items to users, and use temporal visibility to non-uniformly weight missing data for calculating their pre-use preferences. Then, we identify items with low pre-use preference in missing data as uninteresting items and fill them with low values. Finally, the filled rating matrix is directly applied to the existing collaborative filtering algorithms. It is worth noting that pre-use preferences reflect whether users are willing to interact with items, which is essentially an implicit feedback. Our approach actually combines item temporal visibility and implicit feedback to identify the uninteresting items in the missing data, so that both the explicit rating data and the missing data can be fully utilized. The problem that the observed ratings with selection bias can be alleviated since the distribution of rating data conforms to the real preference of users.

3.2. Pre-use preference analysis

Pre-use preferences are generally based on the user's preference for the external features of the item, which can be obtained without actual use, such as the type of film, starring, etc. Post-use preference depends on the user's preference for the internal characteristics of the item, such as film storyline. For items with low pre-use preferences, users will not interact when they see them. It can be inferred that their post-use preference is very low, that is, the rating is very low.

For the items rated by users, it can be simply inferred that their pre-use preference is very high. The challenge is how to accurately infer the pre-use preference of unrated items. This is a one-class problem (Pan et al., 2008), that is to say, the observed ratings are positive samples and lack of negative samples. When implicit feedback lacks negative samples, existing researches have proposed two strategies to deal with it. One is sample-based learning (He & McAuley, 2016), which extracts negative samples from missing data. The other is whole-data-based learning, which regards all missing data as negative samples and give them weight as the confidence that they are negative examples. The former does not need to consider all missing data and has high efficiency, which inevitably has the risk of losing valuable information and poor robustness of the model. The latter uses all the data, with higher coverage but lower efficiency. In order to maintain the robustness of the model, we use whole-data based learning. Weighted Regularized Matrix Factorization (WRMF) takes all missing data as negative samples and assigns small unified weights (Hu, Koren & Volinsky, 2008). It is considered that the unrated items have the same confidence as negative samples, but this is inconsistent with the actual situation. The element-wise alternative least squares(eALS) non-uniformly weighted the unrated items based on the popularity of items (He, Zhang, Kan & Chua, 2016). The more popular items are easy to be seen by users if the user does not interact, the higher the confidence of the negative sample has, and a higher weight should be given. However, these methods do not take into account the differences between users and the impact of time factors on popularity.

Most of these current methods are based on an experience that see-but-not-interact items show users' negative preferences. The more likely an item is to be seen by the user and the user does not interact, the higher the confidence that it is a negative example. The key is how to accurately infer whether the user sees the item. At present, the popular method is to use the popularity of items to infer the visibility of items to users, but the reliability is not high. In fact, users with different activities have different visibility to items, and the popularity of items changes over time (He et al., 2014). We will propose a more fine-grained weighting scheme. At the same time, in order to alleviate the efficiency problem caused by overall data learning, we use eALS to optimize the objective algorithm. eALS provides a new idea for processing weighted matrix factorization to optimize parameters at the element level. It avoids the need of matrix inversion in traditional ALS matrix factorization and improves the efficiency.

3.3. ITV-eALS model

In the current Web 2.0 era, many websites display popular items in their recommendation interface. Generally speaking, popular items are easier to be seen by users. Many researches have shown that inactive users tend to browse popular items, while unpopular items are more likely to be browsed by active users (He et al., 2014). It is worth noting that the popularity of items changes over time, and the display of popular items on the website is also real-time. Only when users are active can popular items really be seen by users. Based on the above point of view, we combines the popularity of items and the sum of user activity during the active period of users to measure the visibility of items to users, and non uniformly weights the unrated items.

Firstly, the original rating matrix $R = (r_{ui})_{m\times n}$ is used to construct the pre use preference matrix $= (p_{ui})_{m\times n}$. m is the number of users and n is the number of items,r_{ui} is user u's rating of item i. According to the analysis in 3.2, the rated items are positive samples, and the pre-use preference is set to 1. All unrated items are regarded as negative samples, and the pre-use preference is set to 0:

$$p_{ui} = \begin{cases} 0, r_{ui} = null \\ 1, r_{ui} \neq null \end{cases} \tag{1}$$

Because the unrated items are not all negative samples, this paper constructs the weight matrix $W = (w_{ui})_{m\times n}$. For rated items, we set their weights $w_{ui} = 1$ represents the confidence that it is a positive sample. For the unrated item, its weight $w_{ui} \in [0,1]$ represents the confidence that it is a negative sample, The higher the weight, the higher the confidence that it is a negative sample.

Define user u's activity α_u is number of ratings for this user,$\alpha_u = \sum_{i=1}^{n} p_{ui}$. User u's rating time for item i is t_{ui}, we define user u's active period is $[t_{min}, t_{max}]$.t_{min} is user u's earliest rating time,t_{max} is user u's latest rating time. We define the popularity of item i β_i is the number of ratings of item i during the user u's active period $[t_{min}, t_{max}]$, $\beta_i = \sum_{u=1}^{m} p_{ui}, t_{ui} \in [t_{min}, t_{max}]$. After smoothing α and β with log function, the maximum value is used for normalization. Smoothing can alleviate the impact of a few extremely active users or popular items:

$$\widehat{\alpha}_u = \frac{log(\alpha_u)}{max(log(\alpha))} \tag{2}$$

$$\widehat{\beta}_i = \frac{log(\beta_i)}{max(log(\beta))} \tag{3}$$

$\widehat{\alpha}_u$ and $\widehat{\beta}_i$ are linearly weighted (weight coefficient $\varepsilon \in [0,1]$).

$$w_{ui} = \begin{cases} 1, p_{ui} = 1 \\ \varepsilon\widehat{\alpha}_u + (1-\varepsilon)\widehat{\beta}_i, p_{ui} = 0 \end{cases} \tag{4}$$

Weight matrix W can be constructed by Eq.(4). We use pre-use preference matrix P and weight matrix W to construct the weighted matrix factorization.Matrix P can be decomposed into two low-rank matrices X and Y. eALS is chosen to optimize the objective function J:

$$J = \sum_{u=1}^{m}\sum_{i=1}^{n} w_{ui}(p_{ui} - \widehat{r}_{ui})^2 + \lambda\left(\sum_{u=1}^{m}||x_u||^2 + \sum_{i=1}^{n}||y_i||^2\right) \tag{5}$$

$X \in R^{k\times m}, Y \in R^{k\times n}$ represent the implicit factor matrix of users and items respectively,k is the number of features. x_u represents the u row of matrix X,y_i represents the i row of matrix Y, they represent latent features vector for user u and item i. $\widehat{r}_{ui} = x_u {y_i}^T$, λ is a regularization parameter. To minimize the loss function J, we first random initialization X and Y, then get the derivative of objective function Eq. (5) with respect to x_{uf}:

$$\frac{\partial J}{\partial x_{uf}} = -2\sum_{i=1}^{n}\left(p_{ui} - \widehat{r}_{ui}^{f}\right)w_{ui}y_{if} + 2x_{uf}\sum_{i=1}^{n} w_{ui}y_{if}^2 + 2\lambda x_{uf} \tag{6}$$

Here$\widehat{r}_{ui}^{f} = \widehat{r}_{ui} - x_{uf}y_{if}$. By setting $\frac{\partial J}{\partial x_{uf}} = 0$, the solution of x_{uf} can be obtained:

$$x_{uf} = \frac{\sum_{i=1}^{n}\left(p_{ui} - \widehat{r}_{ui}^{f}\right)w_{ui}y_{if}}{\sum_{i=1}^{n} w_{ui}y_{if}^2 + \lambda} \tag{7}$$

Similarly, the solution of y_{if} can be obtained:

Table 1
Dataset Statistics.

Dataset	MovieLens 100 k	MovieLens latest	Amazon CDs
#Ratings	100,000	100,836	105,157
#Users	943	610	2588
#Items	1682	9742	2294
Sparsity	93.7 %	98.3 %	98.2 %

$$y_{if} = \frac{\sum_{u=1}^{m}\left(p_{ui} - \widehat{r}_{ui}^{f}\right)w_{ui}x_{uf}}{\sum_{u=1}^{m} w_{ui}x_{uf}^2 + \lambda} \tag{8}$$

The objective function can be optimized by repeating Eq. (7) and Eq. (8) until matrices and converge to a local optimum. Finally, matrix $\widehat{P}$ can be approximated by calculating an inner product of X and Y, $\widehat{P} = XY^T$. Each element $\widehat{p}_{ui}$ in matrix $\widehat{P}$ represents a pre-use preference of user u for item i, $\widehat{p}_{ui} \in [0,1]$, the closer to 1, the higher the pre-use preference.

3.4. Mining and filling of uninteresting items

After calculating the pre-use preferences of unrated items, we can identify items that users are not interested in. Firstly, the θ% items with the lowest pre-use preference is used as a candidate for the uninteresting items, and parameters θ can be adjusted in order to avoid identifying items that users may be interested in as uninteresting item. For each user u, n_u items with the lowest pre-use preference among the candidates can be selected as uninteresting items, where $n_u = ratio * r_u$, r_u is the number of observed ratings by user u. Here *ratio* controls the proportion of uninteresting items to observed ratings. If the ratio setting is too low and the filling quantity is small, the influence of user selection bias may not be fully alleviated. If the ratio is set too high, the rating data may be biased to the user's negative preference. The parameter *ratio* can be adjusted to achieve the best recommendation accuracy.

When the missing item is identified as uninteresting item, a low value will be filled. Because users will not interact with an item with low pre-use preference, filling it with a low value can prevent it from being recommended and alleviate the problem of lack of negative samples due to user selection bias. We adopts a unified low value σ and adjust it to achieve the best recommendation accuracy.

It is worth mentioning that our filling method can be applied to any collaborative filtering algorithm, because it only replaces the original rating matrix with the filled matrix. In other words, this method is orthogonal to all collaborative filtering algorithms and can be easily applied.

4. Experiments

4.1. Experimental setup

We evaluate on three real-world datasets: MovieLens 100 k,[1] MovieLens latest[2] and Amazon CDs.[3] Since the high sparsity of the original datasets makes it difficult to evaluate recommendation algorithms(over half users have only one rating in Amazon CDs).We follow the common practice of MovieLens 100 k to filter out users and items with less than 20 interactions. The statistical data are shown in Table 1.

For topN recommendation, N is the number of items recommended for each user, with values of 5, 10 and 20. For each data set, 80 % of the ratings were randomly selected as training data and the remaining 20 %

[1] https://files.grouplens.org/datasets/movielens/ml-100k.zip.
[2] https://files.grouplens.org/datasets/movielens/ml-latest.zip.
[3] https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/ratings_CDs_and_Vinyl.csv.

Table 2
Rating Distribution.

Dataset	MovieLens 100 k	MovieLens latest	Amazon CD
Low ratings(1 or 2)	17.48 %	13.41 %	8.36 %
High ratings(3、4 or 5)	82.52 %	86.59 %	91.64 %

as test data. Five cross-validations for all experimental results are performed. Only the items with 4 and 5 are considered as relevant items, because it is more meaningful to successfully recommend highly rated items(Cremonesi, Koren & Turrin, 2010). Three metrics are used to measure the accuracy of topN recommendation: precision, recall, normalized discounted cumulative gain(NDCG):

$$Precision = \frac{\sum_{u \in U}|R(u) \cap T(u)|}{\sum_{u \in U}|R(u)|} \tag{9}$$

$$Recall = \frac{\sum_{u \in U}|R(u) \cap T(u)|}{\sum_{u \in U}|T(u)|} \tag{10}$$

$$NDCG@N = \frac{DCG@N}{IDCG@N}\ ; DCG@N = \sum_{i=1}^{N}\frac{2^{rel_i}-1}{log_2(i+1)} \tag{11}$$

Where $R(u)$ is a group of items recommended to user u, and $T(u)$ is the relevant items of user u in the test set. DCG@N is discounted cumulative gain and make the top ranked items in $R(u)$ gain higher, the bottom ranked items are discounted.rel_i indicates whether the item ranked i in $R(u)$ is in $T(u)$. If it exists, then$rel_i = 1$, otherwise$rel_i = 0$. IDCG@N is the value of DCG@N in the ideal case.

4.2. Verification of selection bias and its impact

Firstly, the observed rating data has user selection bias, which is clearly reflected in the three real-world datasets. The rating distribution is shown in Table 2.

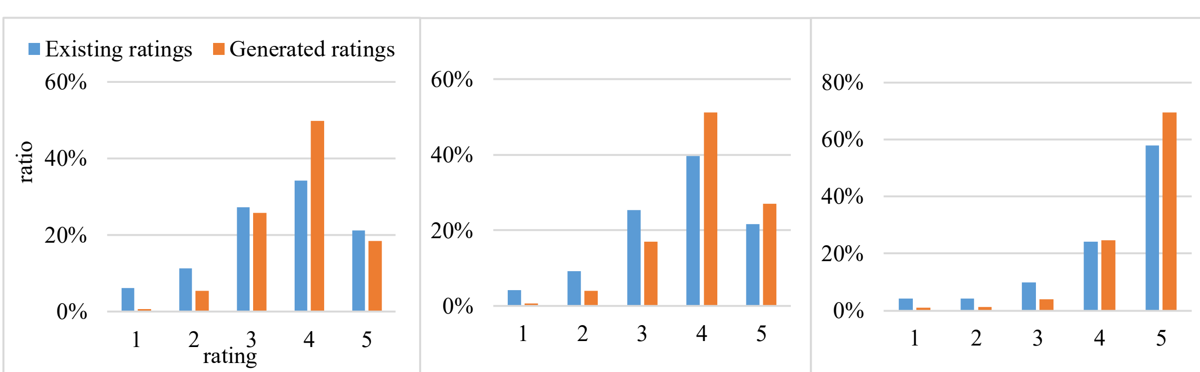
According to Table 2, it can be observed that there are far more high ratings than low ratings. In order to verify its impact on collaborative filtering algorithm for topN recommendation, a group of simple filling experiments are set up. First, for MovieLens 100 k, MovieLens latest and Amazon CDs datasets, we randomly selected 100 unrated items for each user respectively. The first group uses EMDP algorithm to fill in these unrated items, which is a filling algorithm based on observed ratings mentioned in section2, using the same parameter settings as in paper (Ma, King & Lyu, 2007). The second group fills these random unrated items with Zere points(RZF), which is equivalent to randomly selecting uninteresting items among the missing data. It is the simplest implementation of our approach. The experiment uses two most classic and widely used collaborative filtering algorithms ItemCF (Sarwar et al., 2001) and BiasSVD (Koren, Bell & Volinsky, 2009) to compare the impact of the two filling algorithms on topN recommendation before and after filling. The experimental results are shown in Tables 3.

According to Table 3, in the two datasets, RZF simply fills the unrated items with 0, which alleviates the problem of lack of negative samples due to user selection bias, and can significantly enhance the topN recommendation accuracy. EMDP predicts the unrated item rating and fills it according to the observed ratings, and the effect is not obvious or even worse. The rating distribution filled by EMDP algorithm is shown in Fig. 1.

According to Fig. 1, it can be clearly observed that EMDP algorithm fills high ratings for most unrated items. This is consistent with the analysis in the section 2. The filling algorithm based on the observed rating is often affected by the user's selection bias. However, this does not accord with the real rating distribution and exacerbates the impact of user selection bias. It can also explain why its recommendation accuracy is poor. Through this set of comparative experiments, the influence of user selection bias on topN recommendation is sufficiently proved. As the simplest implementation of our approach, RZF also verifies the effectiveness of our approach.

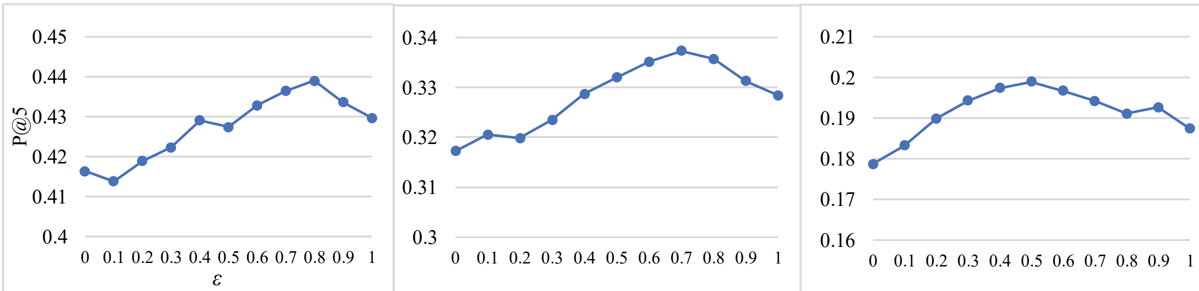
Table 3
Accuracy of two methods.

Datasets	Metrics	ItemCF	BiasSVD	EMDP + ItemCF	EMDP + BiasSVD	RZF+ ItemCF	RZF+ BiasSVD
MovieLens 100 k	P@5	0.0789	0.0815	0.0705	0.0793	**0.2073**	**0.2085**
	R@5	0.0244	0.0274	0.0240	0.0285	**0.1013**	**0.1197**
	NDCG@5	0.0834	0.0878	0.0698	0.0760	**0.2315**	**0.2359**
MovieLens latest	P@5	0.0632	0.0653	0.0259	0.0392	**0.1357**	**0.1424**
	R@5	0.0224	0.0233	0.0221	0.0134	**0.0631**	**0.0727**
	NDCG@5	0.0696	0.0715	0.0355	0.0431	**0.1501**	**0.1635**
Amazon CD	P@5	0.0119	0.0125	0.0186	0.0199	**0.0703**	**0.0751**
	R@5	0.0078	0.0084	0.0139	0.0142	**0.0514**	**0.0547**
	NDCG@5	0.0131	0.0145	0.0207	0.0223	**0.0827**	**0.0883**



(a) MovieLens 100k (b) MovieLens latest (c) Amazon CD

Fig. 1. Rating distributions.



(a) MovieLens 100k (b) MovieLens latest (c) Amazon CD

Fig. 2. Influence of ε.

Table 4
Accuracy of three methods.

DataSets	Metrics	WRMF	eALS	ITV-eALS
MovieLens 100 k	P@5	0.4017	0.4184	**0.4390**
	R@5	0.1471	0.1494	**0.1582**
	NDCG@5	0.4282	0.4468	**0.4697**
Movielens Latest	P@5	0.3159	0.3195	**0.3373**
	R@5	0.0903	0.0932	**0.0970**
	NDCG@5	0.3339	0.3392	**0.3575**
Amazon CDs	P@5	0.1718	0.1846	**0.1989**
	R@5	0.1083	0.1159	**0.1236**
	NDCG@5	0.1943	0.2047	**0.2155**

Table 5
Distribution of pre-use preferences.

pre-use preferences	MovieLens 100 k	MovieLens latest	Amazon CDs
[0,0.2)	92.64 %	97.24 %	96.71 %
[0.2,0.4)	5.40 %	2.03 %	2.73 %
[0.4,0.6)	1.48 %	0.52 %	0.36 %
[0.6,0.8)	0.38 %	0.15 %	0.17 %
[0.8,1]	0.1 %	0.06 %	0.03 %

4.3. Experiment of Pre-use preference model

Two algorithms WRMF (Hu, Koren & Volinsky, 2008) and eALS (He, Zhang, Kan & Chua, 2016) are used as comparison algorithms. They are the same as ITV-eALS. They are all algorithms based on whole data learning to solve the one-class problem of implicit feedback.

WRMF: It is the most classical one-class collaborative filtering method. All unrated items are regarded as negative samples and given a smaller unified weight.

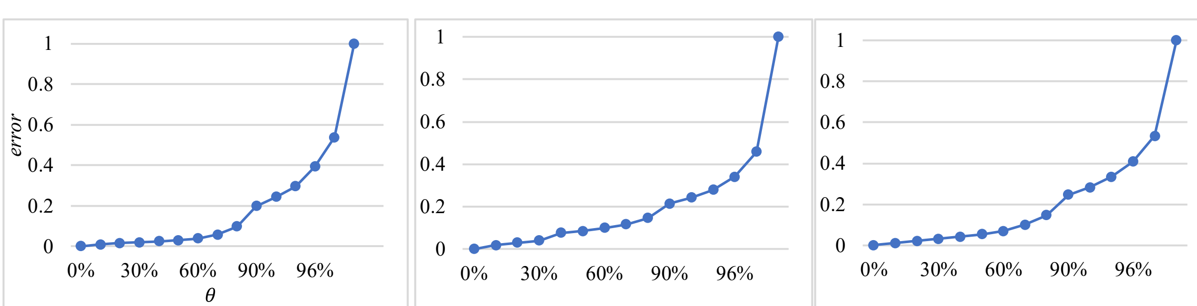
eALS: It is the most advanced implicit matrix decomposition method, which regards all unrated items as negative samples and weights them according to the popularity of items.

It should be noted that due to implicit feedback, different from the setting in 4.1, all ratings in the test set are set as related items, rather than only 4 and 5 as related items. Because they are all implicit matrix factorization models, the number of features k = 20 and regularization parameter$\lambda = 0.01$. The parameters of WRMF and eALS are adjusted according to the method of paper. As for ITV-eALS, ε is weight coefficient of user activity and item popularity,we increase the value range from 0 to 1 in 0.1 increments. The experimental results of recommendation accuracy P@5 is shown in Fig. 2.

According to Fig. 2,we set weight coefficient$\varepsilon = 0.8$ in MovieLens 100 k, set$\varepsilon = 0.7$ in MovieLens latest and set$\varepsilon = 0.5$ in Amazon CDs.The experimental results are shown in Table 4.

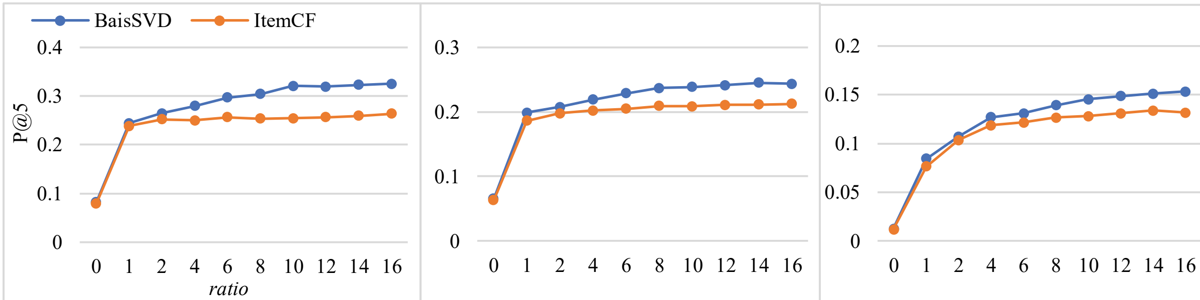
According to Table 4, ITV-eALS can obtain the best recommendation accuracy on all datasets. This shows that combining the user activity and the popularity of items during user active period can more accurately identify the items that see-but-not-interact. Compared with the existing weighting strategy, it realizes the fine-grained weighting of unrated items. Therefore, it can more accurately infer the user's pre-use preference for unrated items, which is very important to identify the uninteresting items in the following work. The range of pre-use preference is [0,1], and the pre-use preference distribution of unrated items is shown in Table 5.

According to Table 5, users have low pre-use preferences for most unrated items, which means users are not interested in most unrated items. This is consistent with the description of related work in section 2. At the same time, this can also verify the reason why EMDP and other filling algorithms only based on existing ratings fill in high rating for most unrated items, but the effect is worse, because it violates the fact that users are not interested in most unrated items and cannot reflect users' real preferences. This also proves the effectiveness of the our approach, because this restores the user's real preferences of users. Compared with the existing ratings with user selection bias, the filled rating matrix can better reflect the real rating distribution of users,



(a) MovieLens 100k (b) MovieLens latest (c) Amazon CD

Fig. 3. Influence of θ.



(a) MovieLens 100k (b) MovieLens latest (c) Amazon CD

Fig. 4. Influence of *ratio*.

Table 6
Influence of σ.

DataSets	Metrics	0	1	2	3	4	5
MovieLens 100 k	BiasSVD	0.3234	**0.3339**	0.3197	0.2595	0.1661	0.0608
	ItemCF	0.2639	0.2821	**0.2852**	0.2684	0.1728	0.0269
Movielens Latest	BiasSVD	0.2452	**0.2519**	0.2486	0.2044	0.1246	0.0432
	ItemCF	0.2127	**0.2230**	0.2194	0.1816	0.1175	0.0278
Amazon CDs	BiasSVD	0.1532	0.1586	**0.1617**	0.1067	0.0384	0.0078
	ItemCF	0.1336	0.1352	**0.1381**	0.0894	0.0357	0.0049

which is helpful for collaborative filtering algorithm to recommend more accurately..

4.4. Experiment of our data filling based on ITV-eALS

4.4.1. Influence of parameters

ItemCF and BiasSVD were used to observe the effect of filling on topN recommendation.

Firstly, an *error* rate is defined to represent the probability of identifying the relevant items in the test set as uninteresting items when setting the parameter θ:

$$error = \frac{\sum_{u \in U}|B(u) \cap T(u)|}{\sum_{u \in U}|T(u)|} \tag{12}$$

where $B(u)$ represents the uninteresting items of user u, $T(u)$ represents the relevant items of user u in the test set.

θ control the candidate items of uninteresting items, increase the value range from 0 % to 90 % in 10 % increments, range from 90 % to 100 % in 2 % increments,and observe the change of *error* at different values. The experimental results are shown in Fig. 3.

According to Fig. 3, *error* rate have the same trend in both datasets, when θ is greater than 90 %,the *error* rate will rise very fast. In other words, it is likely to lead to the identification of items of interest to users as uninteresting items. Therefore, we set$\theta = 90$ %.

Then fixed$\theta = 90$ %, the parameter ratio starts from 0 to 16 and increases in increments of 2. The experimental results of recommendation accuracy P@5 are shown in Fig. 4.

It can be observed that when the *ratio* is between 0 and 16, The P@5 of recommendation show similar trends, and they all increase steadily with the increase of *ratio*. It is well understood that the change is relatively gentle after the *ratio* reaches 10. At this time, enough negative samples have been filled to alleviate the problem of lack of negative samples caused by user selection bias. As the *ratio* increases, uninteresting items will be further excluded from the topN recommendation, which helps to improve accuracy. Considering the above problems, *ratio* can be set as 16.

When$\theta = 90$ %, *ratio*=16, and parameter σ take values from 0 to 5, the experimental results of recommendation accuracy P@5 are shown in Table 6.

According to Table 6, when parameter σ is less than 3, the accuracy P@5 does not change significantly, which indicates that our method is not sensitive to the low value of filling. It can be observed that when σ exceeds 3, accuracy decreases significantly. This is due to the high fill rating, which exacerbates the lack of negative samples. However, the *ratio* is 16 at this time, and most uninteresting items have been excluded from the topN recommendation candidate. Compared the results in Table 3, the accuracy is still better than that before filling. This also proves the effectiveness of the strategy of excluding uninteresting items from the topN recommendation list. It can also be observed that filling an appropriate low value is more effective than filling an extreme rating of 0 (Hwang et al., 2016; Cremonesi, Koren & Turrin, 2010).

4.4.2. Experimental results of data filling based on ITV-eALS

In order to verify the effectiveness of data filling based on ITV-eALS, ItemCF and BiasSVD are used to compare the impact of our approach on topN recommendation before and after filling. At the same time, several classical recommendation algorithms are selected for comparison, and the brief description of each algorithm is as follows:

(1) ItemKNN (Hu, Koren & Volinsky, 2008):It is the most popular variant of ItemCF for topN recommendation, and it is also the popular baseline method at present. We adopt the same setting.
(2) PureSVD (Cremonesi, Koren & Turrin, 2010): It fill all unrated items with 0, and then perform traditional singular value decomposition.
(3) ITV-eALS: The weighted matrix factorization algorithm based on user visibility is used to calculate the pre-use preference of unrated items in this paper.
(4) eALS (He et al., 2017): This is a weighted matrix factorization method for item recommendation, treating all unobserved interactions as negative instances and weighting them non-uniformly by the item popularity.
(5) CDAE (Wu et al., 2016): It uses a denoising autoencoder structure for CF while integrating user-specific latent features.
(6) NeuMF (He et al., 2017): This is a state-of-the-art algorithm of matrix factorization that uses a multi-layer neural network to learn the interaction function between users and items.

Table 7
Accuracy in the MovieLens 100 k.

Metrics	ItemCF	BiasSVD	AutoRec	MLP	ITV-eALS + ItemCF	ITV-eALS + BiasSVD	ITV-eALS + AutoRec	ITV-eALS + MLP	ItemKNN	PureSVD	ITV-eALS
P@5	0.0789	0.0815	0.1071	0.1493	0.2852	0.3339	0.3442	0.228	0.2567	0.2806	0.3041
P@10	0.0766	0.0756	0.0930	0.1382	0.2456	0.2721	0.2967	0.1847	0.2127	0.2261	0.2491
P@20	0.0694	0.0631	0.0808	0.1181	0.1935	0.2103	0.2443	0.1439	0.1643	0.1724	0.1949
R@5	0.0244	0.0274	0.0260	0.0536	0.1492	0.1796	0.1090	0.0697	0.1373	0.1663	0.1760
R@10	0.0496	0.0523	0.0436	0.0901	0.2477	0.2782	0.1800	0.0991	0.2219	0.2560	0.2693
R@20	0.0981	0.0884	0.0775	0.1502	0.3739	0.4157	0.2874	0.1412	0.3286	0.3717	0.3957
NDCG@5	0.0834	0.0878	0.1041	0.161	0.3141	0.3793	0.3653	0.2317	0.2910	0.3238	0.3494
NDCG@10	0.0843	0.0866	0.0979	0.1604	0.3152	0.3688	0.3469	0.2091	0.2896	0.3183	0.3443
NDCG@20	0.0931	0.0896	0.0977	0.167	0.3326	0.3863	0.3465	0.1947	0.3011	0.3364	0.3649

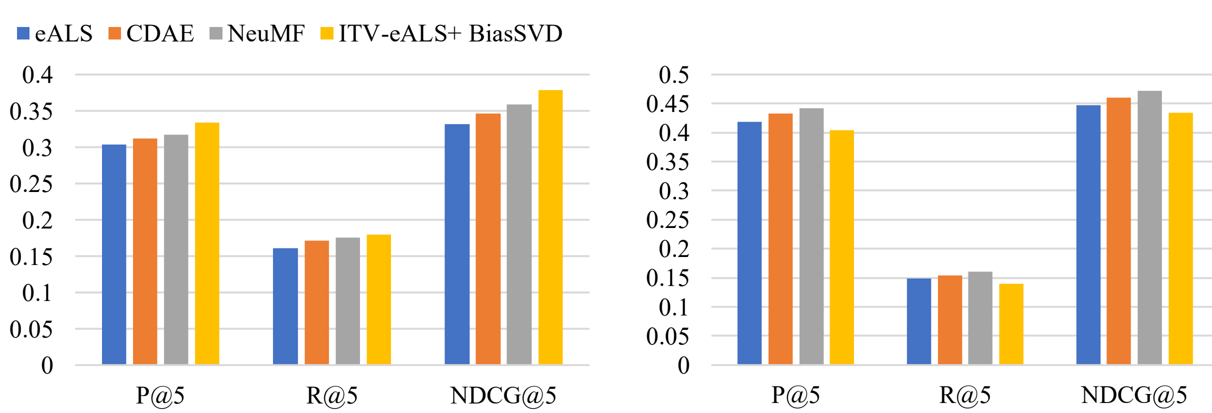
Table 8
Accuracy in the MovieLens latest.

Metrics	ItemCF	BiasSVD	AutoRec	MLP	ITV-eALS + ItemCF	ITV-eALS + BiasSVD	ITV-eALS + AutoRec	ITV-eALS + MLP	ItemKNN	PureSVD	ITV-eALS
P@5	0.0632	0.0653	0.1018	0.0828	0.2230	0.2519	0.2302	0.1038	0.1986	0.2324	0.2355
P@10	0.0521	0.0546	0.0888	0.0601	0.1886	0.2012	0.2030	0.0941	0.1629	0.1842	0.1947
P@20	0.0439	0.0447	0.0672	0.0443	0.1486	0.1625	0.1740	0.0893	0.1283	0.1431	0.1495
R@5	0.0224	0.0233	0.0219	0.0156	0.1027	0.1231	0.0628	0.0196	0.094	0.1103	0.1240
R@10	0.0353	0.0347	0.0403	0.0218	0.1620	0.1827	0.1090	0.0326	0.1537	0.1634	0.1860
R@20	0.0586	0.0590	0.0592	0.0295	0.2414	0.2738	0.1806	0.0694	0.2323	0.2334	0.2709
NDCG@5	0.0696	0.0715	0.1065	0.0822	0.2423	0.2807	0.2343	0.1128	0.2191	0.2596	0.2676
NDCG@10	0.0642	0.0663	0.0963	0.0681	0.2343	0.2778	0.2254	0.1055	0.2141	0.2436	0.2609
NDCG@20	0.0665	0.0670	0.0878	0.0577	0.2402	0.2765	0.2298	0.1096	0.2247	0.2461	0.2694

Table 9
Accuracy in the Amazon CDs.

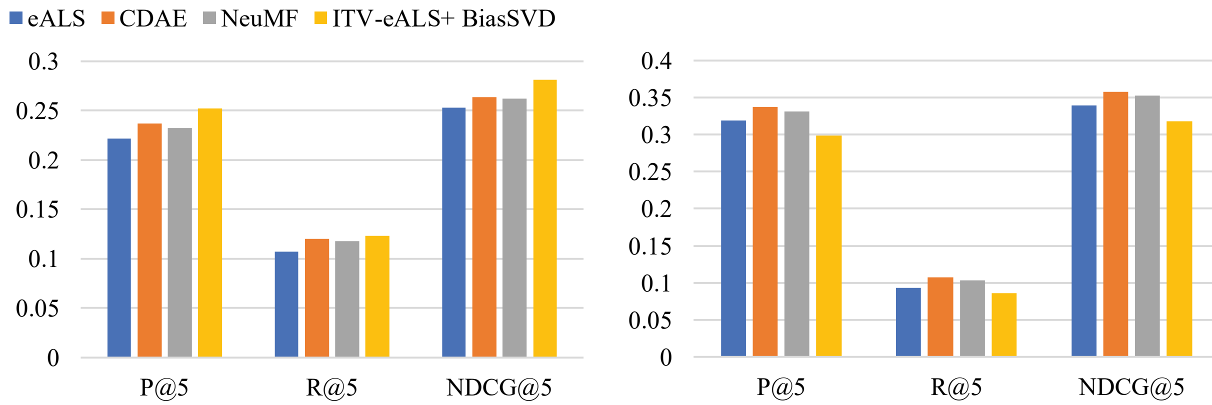
Metrics	ItemCF	BiasSVD	AutoRec	MLP	ITV-eALS + ItemCF	ITV-eALS + BiasSVD	ITV-eALS + AutoRec	ITV-eALS + MLP	ItemKNN	PureSVD	ITV-eALS
P@5	0.0119	0.0125	0.0139	0.0478	0.1381	0.1617	0.1398	0.0565	0.1178	0.1346	0.1452
P@10	0.0092	0.0097	0.0105	0.0423	0.1086	0.1263	0.1103	0.0467	0.0918	0.1043	0.1158
P@20	0.0069	0.0074	0.0095	0.0341	0.0839	0.0981	0.0855	0.0355	0.0667	0.0759	0.0853
R@5	0.0078	0.0084	0.0082	0.0333	0.1072	0.1227	0.0908	0.0353	0.0906	0.1008	0.1136
R@10	0.0120	0.0133	0.0122	0.0580	0.1639	0.1860	0.1437	0.0598	0.1382	0.1530	0.1745
R@20	0.0177	0.0202	0.0248	0.0934	0.2515	0.2679	0.2185	0.0924	0.1998	0.2221	0.2562
NDCG@5	0.0131	0.0145	0.0159	0.0503	0.1662	0.1908	0.1581	0.0609	0.1421	0.1581	0.1749
NDCG@10	0.0132	0.0149	0.0145	0.0522	0.1760	0.1975	0.157	0.0624	0.1448	0.1609	0.1804
NDCG@20	0.0147	0.0171	0.019	0.0643	0.1971	0.2232	0.1806	0.0728	0.1644	0.1832	0.2077

According to Table 7, Table 8 and Table 9, only considering the existing rating data, the accuracy of ItemCF, BiasSVD and AutoRec is not ideal. After filling with our approach, the recommendation accuracy of ItemCF, BiasSVD and AutoRec is significantly improved. This fully shows that the effective use of users' negative preferences implied by missing data plays a positive role in topN recommendation. ItemKNN recommends items with high similarity to their rated items for users, and sorts them by calculating non standardized preferences, avoiding the



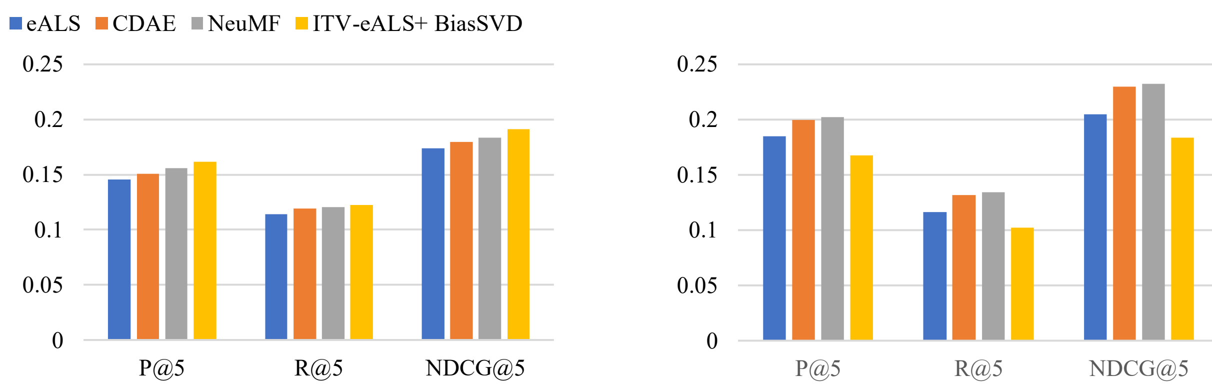
(a) Only high ratings are considered (b) All ratings are considered

Fig. 5. Results of MovieLens 100 k.



(a) Only high ratings are considered (b) All ratings are considered

Fig. 6. Results of MovieLens latest.



(a) Only high ratings are considered (b) All ratings are considered

Fig. 7. Results of Amazon CDs.

bias of ranking by rating prediction. Therefore,it shows better accuracy than ItemCF. However, it only considers the rated data. ITV-eALS + ItemCF makes full use of the missing data, alleviates the user's selection bias, so the recommendation accuracy is better than ItemKNN.

Comparing PureSVD with RZF algorithm in Tables 3, our approach also shows better recommendation accuracy. Because PureSVD simply takes all unrated items as uninteresting items, RZF randomly selects uninteresting items from unrated items, and it does not really identify uninteresting items in missing data. This shows the effectiveness of accurately identifying uninteresting items.

As one of the state-of-the-art recommendation algorithms based on implicit feedback, ITV-eALS takes into account both observed data and missing data, makes full use of the negative preference implied by missing data. Therefore, it shows quite high accuracy but is still below to ITV-eALS + BiasSVD. However, implicit feedback is different from explicit feedback, which can not clearly express user preference and its degree.

In order to further explore the difference between implicit feedback method and explicit feedback method in topN recommendation, we use ITV-eALS + BiasSVD to compare with the current popular recommendation algorithm based on implicit feedback. Two groups of experiments are set up. The first group adopts the same setting as previous experiments, only the items with ratings of 4 and 5 in the test set are considered as related items in order to observe the recommendation effect of the algorithm on the items with high rating. The second group takes all items in the test set as related items, that is, it only considers whether users interact with items, and does not care whether they give high or low rating after interaction. The experimental results are shown in Fig. 5, Fig. 6 and Fig. 7.

According to Fig. 5, Fig. 6 and Fig. 7, ITV-eALS + BiasSVD shows similar accuracy to the currently popular topN recommendation algorithm based on implicit feedback in both groups of experiments. The classical recommendation algorithm based on explicit feedback can also show good recommendation accuracy after eliminating the influence of selection bias.

In the first group of experiments, when only the items with high rating in the test set are considered, the accuracy of ITV-eALS + BiasSVD is better than other methods based on implicit feedback which shows the limitation of implicit feedback. Implicit feedback is not clear about the expression of user preferences, and can not express the degree of preferences. In other words, the implicit feedback method only considers the pre-use preferences. However, a high pre-use preference does not necessarily mean a high post-use preference. This may lead to recommend items that will not have high rating after actual interaction.

In a word, the method based on implicit feedback is more inclined to recommend the items that users may interact with, but it can not guarantee whether users will like it after interaction, which is consistent with the results of the second group of experiments. The implicit feedback methods have better accuracy when all the ratings of the test set are considered. However, there is obviously a risk of recommending items that users do not like after actual interaction and resulting in worse user experience.

We use the data filling based on ITV-eALS accurately identify the uninteresting items in the missing data and inject them into the existing ratings, which helps explicit feedback methods overcome the problem of user selection bias. In other words, we fully integrate the advantages of explicit feedback and implicit feedback, so it can make more reliable recommendations to ensure that users are satisfied before and after purchase. As the experimental results show, although it is the basic algorithm, it shows better accuracy than other comparison algorithms under the enhancement of our approach.

5. Conclusion

We analyze and demonstrate the impact of user selection bias on topN recommendation in explicit rating data and further analyze the limitations of existing methods that only consider the observed ratings. Based on these analysis, we propose a general data filling strategy based on uninteresting items, which makes full use of the negative preference implied by missing data. In order to mine uninteresting items in missing data, the concept of pre-use preference is proposed. According to the experience that see-but-not-interact items show users' negative preferences, we combined with user activity, item popularity and time factors to measure users' visibility to the items. Based on this, ITV-eALS algorithm is proposed to model the pre-use preferences of missing data. Items with low pre-use preference are regarded as uninteresting items and filled with low values. Through comprehensive experiments, we demonstrated that our approach can effectively improve the accuracy of collaborative filtering algorithm without using any auxiliary information,and our results show that the explicit feedback method shows better recommendation performance than the implicit feedback method after alleviating the selection bias. Furthermore, our approach is orthogonal to various existing collaborative filtering algorithms, so it can be easily applied.

CRediT authorship contribution statement

Lei Shi: Software, Writing – original draft, Visualization. **Shuqing Li:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration. **Xiaowei Ding:** Validation, Formal analysis. **Zhan Bu:** Data curation, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by Chinese National Social Science Foundation (17BTQ028), Major Natural Science Research Projects of Colleges and Universities in Jiangsu Province of China (19KJA510011), Graduate Research and Practice Innovation Program in Jiangsu Province of China (KYCX20_1348), National Natural Science Foundation of China (71871109).

References

Ashokan, A., & Haas, C. (2021). Fairness metrics and bias mitigation strategies for rating predictions. *Information Processing & Management, 58*(5), Article 102646.

Chae, D. K., Kang, J. S., Kim, S. W., & Choi, J. (2019). May). Rating augmentation with generative adversarial networks towards accurate collaborative filtering. In *In The World Wide Web Conference* (pp. 2616–2622).

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., & He, X. (2020). Bias and debias in recommender system: A survey and future directions. arXiv preprint arXiv: 2010.03240.

Cremonesi, P., Koren, Y., & Turrin, R. (2010). September). Performance of recommender algorithms on top-n recommendation tasks. In *In Proceedings of the fourth ACM conference on Recommender systems* (pp. 39–46).

Cheng, X., Feng, L., & Gui, Q. (2019). Collaborative Filtering Algorithm based on Data Mixing and Filtering. *International Journal of Performability Engineering, 15*(8), 2267.

Chen, Y. L., Yeh, Y. H., & Ma, M. R. (2021). A movie recommendation method based on users' positive and negative profiles. *Information Processing & Management, 58*(3), Article 102531.

Da Silva, J. F. G., De Moura Junior, N. N., & Caloba, L. P. (2018). In *July). Effects of data sparsity on recommender systems based on collaborative filtering* (pp. 1–8). IEEE.

Deldjoo, Y., Bellogin, A., & Di Noia, T. (2021). Explaining recommender systems fairness and accuracy through the lens of data characteristics. *Information Processing & Management, 58*(5), Article 102662.

Elahi, M., Kholgh, D. K., Kiarostami, M. S., Saghari, S., Rad, S. P., & Tkalčič, M. (2021). Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management, 58*(5), Article 102655.

He, R., & McAuley, J. (2016, February). VBPR: visual bayesian personalized ranking from implicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

Ha, T., & Lee, S. (2017). Item-network-based collaborative filtering: A personalized recommendation method based on a user's item network. *Information Processing & Management, 53*(5), 1171–1184.

Hwang, W. S., Parc, J., Kim, S. W., Lee, J., & Lee, D. (2016). In *May). "Told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering* (pp. 349–360). IEEE.

He, X., Gao, M., Kan, M. Y., Liu, Y., & Sugiyama, K. (2014). July). Predicting the popularity of web 2.0 items based on user comments. In *In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 233–242).

He, X., Zhang, H., Kan, M. Y., & Chua, T. S. (2016). July). Fast matrix factorization for online recommendation with implicit feedback. In *In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 549–558).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). April). Neural collaborative filtering. In *In Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Hu, Y., Koren, Y., & Volinsky, C. (2008). In *December). Collaborative filtering for implicit feedback datasets* (pp. 263–272). Ieee.

Jawaheer, G., Szomszor, M., & Kostkova, P. (2010). September). Comparison of implicit and explicit feedback from an online music recommendation service. In *In proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems* (pp. 47–51).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37.

Liang, D., Charlin, L., McInerney, J., & Blei, D. M. (2016). April). Modeling user exposure in recommendation. In *In Proceedings of the 25th international conference on World Wide Web* (pp. 951–961).

Liao, X., Koch, P., Huang, S., & Xu, Y. (2021). August). Efficient Collaborative Filtering via Data Augmentation and Step-size Optimization. In *In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 1006–1016).

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems, 74*, 12–32.

Marlin, B. M., Zemel, R. S., Roweis, S., & Slaney, M. (2007). July). Collaborative filtering and the missing at random assumption. In *In Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence* (pp. 267–275).

Marlin, B. M., & Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems* (pp. 5–12).

Ma, H., King, I., & Lyu, M. R. (2007). July). Effective missing data prediction for collaborative filtering. In *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 39–46).

Misztal-Radecka, J., & Indurkhya, B. (2021). Bias-Aware Hierarchical Clustering for detecting the discriminated groups of users in recommendation systems. *Information Processing & Management, 58*(3), Article 102519.

Nguyen, M., Yu, J., Nguyen, T., & Yongchareon, S. (2022). High-order autoencoder with data augmentation for collaborative filtering. *Knowledge-Based Systems, 240*, Article 107773.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., & Yang, Q. (2008). One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 502–511). IEEE.

Pan, W., Cui, S., Wen, H., Chen, K., Zhang, C., & Wang, F. (2021). Correcting the User Feedback-Loop Bias for Recommendation Systems. arXiv preprint arXiv:2109.06037.

Ren, Y., Li, G., Zhang, J., & Zhou, W. (2012). October). The efficient imputation method for neighborhood-based collaborative filtering. In *In Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 684–693).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). April). Item-based collaborative filtering recommendation algorithms. In *In Proceedings of the 10th international conference on World Wide Web* (pp. 285–295).

Steck, H. (2010). July). Training and testing of recommender systems on data missing not at random. In *In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 713–722).

Steck, H. (2013). Evaluation of recommendations: Rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 213–220). ACM.

Sarkar, S., Mitsui, M., Liu, J., & Shah, C. (2020). Implicit information need as explicit problems, help, and behavioral signals. *Information Processing & Management, 57*(2), Article 102069.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016). February). Collaborative denoising auto-encoders for top-n recommender systems. In *In Proceedings of the ninth ACM international conference on web search and data mining* (pp. 153–162).

Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017). August). Deep Matrix Factorization Models for Recommender Systems. *In IJCAI, 17*, 3203–3209.

Yuan, X., Han, L., Qian, S., Zhu, L., Zhu, J., & Yan, H. (2021). Preliminary data-based matrix factorization approach for recommendation. *Information Processing & Management, 58*(1), Article 102384.