# BPRH : bayesian personalized ranking for heterogeneous implicit feedback

Qiu, Huihuai; Liu, Yun; Guo, Guibing; Sun, Zhu; Zhang, Jie; Nguyen, Hai Thanh

2018

Qiu, H., Liu, Y., Guo, G., Sun, Z., Zhang, J., & Nguyen, H. T. (2018). BPRH: Bayesian personalized ranking for heterogeneous implicit feedback. Information Sciences, 453, 80-98. doi:10.1016/j.ins.2018.04.027

https://hdl.handle.net/10356/85173

https://doi.org/10.1016/j.ins.2018.04.027

# BPRH: Bayesian Personalized Ranking for Heterogeneous Implicit Feedback

Huihuai Qiu[a,b], Yun Liu[a,*], Guibing Guo[c], Zhu Sun[b], Jie Zhang[b], Hai Thanh Nguyen[d,e]

[a] *School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China*
[b] *Nanyang Technological University, Singapore*
[c] *Northeastern University, Shenyang, China*
[d] *Telenor Research, Oslo, Norway*
[e] *Department of Computer Science, Norwegian University of Science and Technology, Trondheim Norway*

**Abstract**

Personalized recommendation for online service systems aims to predict potential demand by analysing user preference. User preference can be inferred from heterogeneous implicit feedback (i.e. various user actions) especially when explicit feedback (i.e. ratings) is not available. However, most methods either merely focus on homogeneous implicit feedback (i.e. target action), e.g., *purchase* in shopping websites and *forward* in Twitter, or dispose heterogeneous implicit feedback without the investigation of its speciality. In this paper, we adopt two typical actions in online service systems, i.e., *view* and *like*, as auxiliary feedback to enhance recommendation performance, whereby we propose a Bayesian personalized ranking method for heterogeneous implicit feedback (BPRH). Specifically, items are first classified into different types according to the actions they received. Then by analysing the co-occurrence of different types of actions, which is one of the fundamental speciality of heterogeneous implicit feedback systems, we quantify their correlations, based on which the difference of users' preference among different types of items is investigated. An adaptive sampling strategy is also proposed to tackle the unbalanced correlation

*Corresponding author
*Email address:* liuyun@bjtu.edu.cn (Yun Liu )

among different actions. Extensive experimentation on three real-world datasets demonstrates that our approach significantly outperforms state-of-the-art algorithms.
*Keywords:* Recommendation, Heterogeneous Implicit Feedback, Personalized Ranking, Co-occurrence

## 1. Introduction

Recommender systems [1] have been successfully applied in online systems to provide personalized service and thus resolve the *information overload* problem [2, 3]. Generally, recommendation methods are often based on either explicit feedback or implicit feedback. The former methods leverage users' ratings or reviews to predict users' potential interest [4, 5, 6, 7, 8], while the latter exploits users' actions (e.g., purchase, forward, view, like) to estimate user preference [9, 10, 11]. In real-world applications, since explicit feedback is not always available, the algorithms based on implicit feedback [12, 13, 14, 15] have received much attention recently.

Implicit feedback is heterogeneous in real-world systems. A common example shown in Figure 1 is an online shopping website, where users may perform various actions on items such as *view* (browse the details), *like* (click the 'like' button) and *purchase* (buy the item). Some researchers have taken preliminary steps to study the value of implicit feedback [16, 17]. In this paper, we term the feedback to predict as *target action*, which is generally the most concerned feedback for recommendation service providers. For instance, *purchase* is the *target action* in e-commerce systems. Other types of actions (e.g., *view* and *like*) offer side information to help model user preference, called *auxiliary action*. Both *target* and *auxiliary* actions indicate users' positive attitude on items. Generally, *target action* provides stronger indication of user preference than *auxiliary action* [18]. However, in practice, *target action* is scarce while *auxiliary action* is rich. Note that although not available in many scenarios, there is another kind of actions besides *target action* and *auxiliary action*, namely, *negative action*

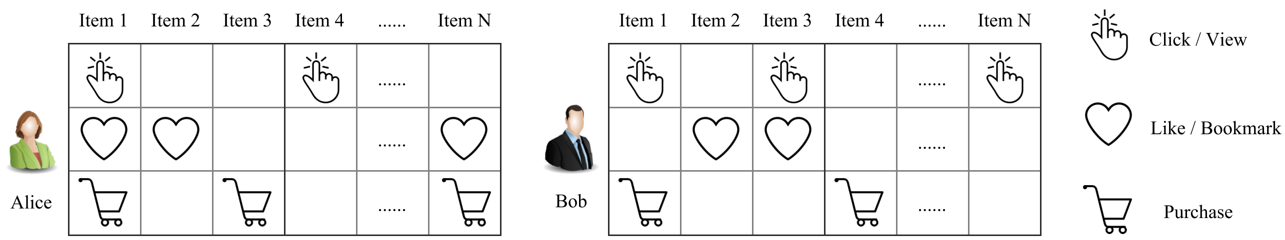



Figure 1: An example of users performing various actions on items in online websites.

(e.g., delete the unwanted items). It implies users' explicit negative attitude on items, thus provides the counter indications about user preference.

Many existing studies [19, 20, 21, 13, 22, 23] merely consider *target action* (homogeneous implicit feedback) and thus inherently suffer from the data sparsity problem [24]. Therefore, some researchers attempt to exploit heterogeneous implicit feedback, and show that both target and auxiliary actions can be integrated for better recommendation [18, 25, 26, 27, 28]. However, these approaches did not pay sufficient attention to the correlation between *target action* and *auxiliary action*. Some works are basically simple combination of submodels that are designed for homogeneous implicit feedback [25, 26], while some studies are merely based on intuitive assumptions which are invalid in some scenarios[13, 18]. The limited improvements achieved by these approaches indicate that the value of *auxiliary action* requires further investigation.

In this paper, we propose a novel Bayesian personalized ranking approach for heterogeneous implicit feedback (BPRH). It integrates multiple types of *auxiliary action* and *target action* into a unified model for better recommendation. In the model, we uncover the essential speciality of heterogeneous implicit feedback and investigate the correlation between *auxiliary action* and *target action*. The main contributions of this study are summarized as follows:

- We propose a *series pairwise assumption* for heterogeneous implicit feedback. Items are classified into three types: with *target action*, with only *auxiliary action* and with no action. We assume that user preference towards items with only *auxiliary action* is weaker than those with *target action*, but stronger than those with no action (*target* $\succ$ *auxiliary* $\succ$ no action), called *series pairwise assumption*. It can be easily extended if *negative action* is available.

- We design a Bayesian Personalized Ranking model for Heterogeneous implicit feedback (BPRH[1]). Specifically, we investigate the fine-granular user preference difference between items with *target action* and those with *auxiliary action*, which could be reflected by the correlation between the two types of actions. Table 1 exhibits the examples of co-occurrence between *target action* and *auxiliary action* in three real-word datasets. Hence, we leverage the co-occurrence to quantify the correlation. A penalty weight parameter $\alpha$ is introduced to control the user preference difference between items with *target action* and *auxiliary action*. To the authors' best knowledge, our BPRH is the first model that uncovers and utilizes the co-occurrence of different types of actions in the case of heterogeneous implicit feedback for effective recommendation.

- We obtain an interesting observation that correlation level between different actions can be counterintuitive. As we consider the two types of *auxiliary action*, intuitively, *like* should be correlated more with *purchase* than *view*. However, this proposition is invalid in the scenario where social networks are attached, e.g. an shopping application where users can follow each other. A possible explanation is that in a connected social network, the motivation for users to click 'like' is more complicated. That

[1]A preliminary report of our work was published at ACM UMAP'16 as an extended abstract[29]. We have extended it in the following aspects. For the techniques, (1) we extend our assumption by considering item-set preference and negative feedbacks; (2) we propose a novel method to build item-set instead of random generation; (3) we design an adaptive sampling strategy for effective model learning. Besides, we add more descriptions and relevant research in the related work, and include new experimental results as well as analysis (i.e., four additional state-of-the-art methods, two more real-world datasets).

Table 1: The co-occurrence of *auxiliary action* and *target action*: the column '*view* & *target action*' (respectively '*like* & *target action*') shows the percentage that *view* (*like*) co-occurs with *target action* w.r.t. the total number of *target action*.

| Datasets | *view* & *target action* | *like* & *target action* | *target action* |
|---|---|---|---|
| Sobazaar | 88% | 32% | *purchase* |
| Tmall | 77% | 31% | *purchase* |
| Xing | 99% | 36% | *reply* |

is to say, there is much noise in *like*. To tackle this issue, we hence filter out those *like* which are less relevant to *purchase*.

- We design a effective learning algorithm which is novel in two aspects. 1) Inspired by the idea in [30] that modeling user preference over item-sets is more effective than over a single item, we propose a novel scheme to build item-sets instead of random generation, thus to further improve recommendation performance. 2) To dispose the unbalanced correlation between *auxiliary action* and *target action*, we propose an adaptive sampling strategy to reduce the sample frequency of items with *auxiliary action* that is less correlated with *target action*. One notable advantage of our adaptive sampling strategy over the existing schemes is that no hyper-parameters is required. Besides, it managed to handle the counterintuitive case mentioned above smoothly.

- Lastly, we conduct comprehensive experiments to evaluate the effectiveness of our proposed model. The empirical studies on three real-world datasets, i.e., Sobazaar, Tmall and Xing, demonstrate that our approach significantly outperforms the state-of-the-art algorithms.

The rest of this article is organized as follows. Section 2 gives an overview of the related research. Our proposed approach is elaborated in Section 3. Section 4 presents the optimization of the proposed method. In Section 5, empirical studies on three real-world datasets are conducted to verify the effectiveness of our method. Finally, Section 6 concludes our work and outlines future research.

## 2. Related Work

This section provides a brief review of related works on implicit feedback for personalized recommendation. They are classified into two types: methods for homogeneous implicit feedback and methods for heterogeneous implicit feedback.

### 2.1. Homogeneous Implicit Feedback

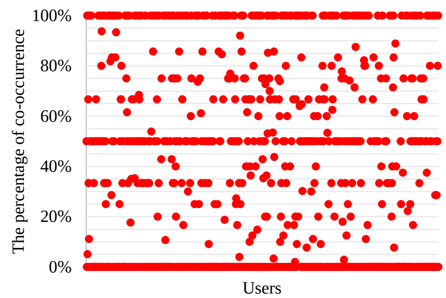
Generally, two types of approaches for homogeneous implicit feedback are widely investigated: pointwise approaches and pairwise approaches. Pointwise approaches take implicit feedback as an absolute score of user preference. For example, to deal with the ambiguity arising in the interpretation of unobserved items (i.e., items with no actions), Pan et al. [31] propose two frameworks to balance the strategies between unobserved items as negative items and as unknown. Hu et al. [20] identify the unique characteristics of implicit feedback and devise a weighted regularized matrix factorization method by introducing confidence weights. Although these works aim at the item prediction task of personalized ranking, none of them is directly optimized for ranking [23]. As a consequence, the recommendation performance is inferior to that of the pairwise approaches [23, 32].

Contrarily, pairwise approaches take implicit feedback as a relative score of user preference. Rendle et al. [23] pioneer this idea by designing a Bayesian personalized ranking (BPR) framework, where they introduce a pairwise assumption that a user prefers an observed item (i.e., item with *target action*) over an unobserved one. After that, a number of variants based on BPR have been proposed to date. For instance, Pan et al. [30] generalize the above assumption by using the relationship between two item-sets instead of two items to reduce the negative impact caused by some counter-examples (e.g., a user may dislike a purchased item or like an item not purchased yet). We follow this idea to propose our assumption while take one step further by designing a novel scheme to build item-sets instead of random pick as in [30]. Later, Pan et al [22] point out that the fundamental assumptions made in BPR: (1) *individual pairwise preference over two items* and (2) *independence between two users*, may not always hold. As a response, they propose an improved *group preference* to relax the *individual* and *independence* assumptions. Zhao et al. [32] introduce social connections information, and estimate user preference based on that of their social friends. Compared with pointwise ones, pairwise approaches are more efficient in item recommendation, as they directly model users' ranking-related preferences. However, all the methods mentioned above cannot accommodate heterogeneous implicit feedback.

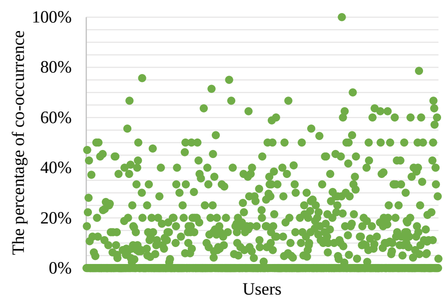
### *2.2. Heterogeneous Implicit Feedback*

The existing studies on heterogeneous implicit feedback also consist of pointwise and pairwise methods. We first make a brief review of the pointwise methods. Li et al. [33] extend the method in [20] by introducing the click-through (i.e., *view*) records to help generate more accurate confidence weights. This work verifies the usefulness of *auxiliary action* for effective recommendation. The model proposed by Pan et al. [5] also demonstrates that *purchase* and *view* can be accommodated to provide more accurate rating prediction. Later, Pan et al. [25] also propose a transfer learning model that integrates *view* into a factorized similarity model [34] via a joint similarity learning method. As pointwise methods, all the above methods suffer from the fundamental shortages as discussed in the previous subsection.

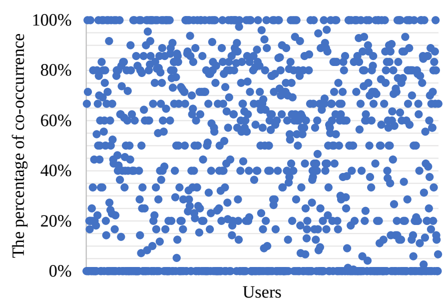
There are also several pairwise approaches for heterogeneous implicit feedback. Pan et al. [26] build two separate BPR models: 1) items with *target action* and without *target action*; 2) items with *auxiliary action* and with no action. The two BPR models are then linearly combined. However, it fails to directly optimize user preference towards items with different types of actions. Lerche et al. [21] try to measure the pairwise preference difference of two items that are both obtained actions from a same user. They mention many possible issues in BPR and design a generic model, i.e., BPR++, to roughly deal with them. One of the issues pointed out is that different types of actions should indicate




(a) Sobazaar




(b) Tmall




(c) Xing

Figure 2: The co-occurance percentage of *like* and *target action* performed by randomly picked users in three real-world dataset.

different user preference levels. Unfortunately, they fail to give further analysis and no specific solution or experiment is presented. In their final model, the recency and the number of obtained actions are considered to modify the pairwise assumption respectively. However, it relies on additional information (i.e., time stamp for each action), without which it performs even worse than BPR [21]. Loni et al. [18] contend that there is a prescribed order (that can be deduced by experience based on the background of the datasets) of different types of actions according to the preference levels behind them. However, it is not easy to determine the accurate preference levels by experience. As users perform actions in different patterns (Figure 2 depicts the co-occurrence percentage of *like* and *target action* of 1,000 randomly picked users), the preference level order should be personalized. Besides, both [21] and [18] devise non-uniform sampling strategies to ameliorate model learning. In [21], a biased sampling is proposed to control the probability of sampling a pair of items that both obtained actions. In [18], manual-setting weights are utilized to determine the sampling frequency of items with different types of actions. Both schemes are severely dependent on manually chosen parameters.

Compared with the approaches mentioned above, our BPRH model has several essential advantages: 1) It is built upon a more unified assumption, i.e., the *series pairwise assumption*, which is able to avoid inconsistent or offset model training caused by separated pairwise assumptions. 2) It takes advantages of the co-occurrence of different types of actions, which is a major characteristics of heterogeneous implicit feedback system, to quantify the user-specific correlation between *target action* and *auxiliary action*[2]. 3) A novel learning algorithm is designed with two considerations: a) It generates item-sets to reduce the chance that items with inconsistent user preference may appear in one same item-set; b) It adopts an adaptive sampling strategy without hyperparameters to handle the unbalance correlation between *target action* and different *auxiliary action*, especially for the counterintuitive case caused by *like* in the scenario where social network is connected.

[2]Note that we study the co-occurrence of *auxiliary action* and *target action* without considering the sequence of their occurrence, which will be left for future work. To clarify, when we talk about two actions co-occurring, either of them could occur first and then be followed by another.

Note that all the related works mentioned above aim at users and their preferences, including long and short-term preferences, which are learned from implicit feedback or user actions. We regard them as user-based methods. Another type of studies is designed for the scenario where user ID is not available, called session-based methods. These methods focus on implicit feedback in a session. Their concern is merely the short-term patterns and the main task is to predict unknown actions within the session. In this paper, we focus on the study of user-based implicit feedback.

## 3. The Proposed BPRH Approach

In this section, we will present the proposed BPRH method that incorporates the influence of heterogeneous implicit feedback into a unified BPR model. We first introduce our *sequential pairwise assumption* by analyzing the drawbacks of user preference assumptions in the existing methods. Then, we fuse the *series pairwise assumption* into the BPR model to further enhance recommendation performance, whereby a simple yet effective method BPRH is proposed.

For clarity, we introduce a number of notations summarized in Table 2 to describe the studied recommendation problem. Let $u \in U, i \in I$ denote user $u$ and item $i$, respectively; $U, I$ denote the user and item set, respectively; $I^u$ is the set of items that user $u$ performs actions on; $I_t, I_{nt}$ are the sets of items with *target action* and without *target action*, respectively; $I_a$ is the set of items with *auxiliary action*, including *view* $I_v$ and *like* $I_l$; Note that $I_t^u \cap I_a^u \neq \emptyset, I_l^u \cap I_v^u \neq \emptyset$, since one user can perform various actions on a same item; $I_{oa}$ is the set of items with only *auxiliary action*, namely, $I_{oa}^u \subseteq I_a^u$, $I_t^u \cap I_{oa}^u = \emptyset$; $I_d$ is the set of items with *delete* $I_n$ is the set of items with no actions.

For each user $u \in U$, we aim to provide a personalized ranking list of items that she has not purchased yet, i.e., $i \in I_{nt}^u$, based on her history behaviour records, i.e., items that she performs actions on $I^u = I_t^u \cup I_a^u \cup I_d^u$.

Table 2: Mathematical notations

| **Notations** | **Description** |
|---|---|
| $u, i/j/k/g$ | user $u$ and items $i/j/k/g$ respectively |
| $U, I$ | the user set and the item set, respectively |
| $I^u$ | the set of items that user $u$ performs actions on |
| $I_t, I_{nt}$ | the sets of items with *target action* and without *target action*, respectively. $I_t \cup I_{nt} = I$ |
| $I_a$ | the set of items with *auxiliary action* |
| $I_{oa}$ | the set of items with only *auxiliary action* |
| $I_v, I_l$ | the set of items with *view*, *like*, respectively. $I_v \cup I_l = I_a$ |
| $I_d$ | the set of items with *delete* (i.e., *negative action*) |
| $I_n$ | the set of items without any action |
| $I_r^u$ | the set of items that are recommended to user $u$ |
| $Pref(u, i)$ | user $u$'s preference to item $i$ |
| $\hat{r}_{ui}$ | the estimated preference of user $u$ to item $i$ |
| $\hat{r}_{u\mathcal{I}}$ | the estimated preference of user $u$ to item-set $\mathcal{I}$ |
| $C^u$ | the *auxiliary-target correlation* for user $u$ |
| $T(u, i)$, $A(u, i)$, $N(u, i)$ | the dataset of *target action*, *auxiliary action* and *negative action*. |

### 3.1. The Series Pairwise Assumption

As mentioned in Section 2, there are mainly two types of assumptions proposed to model user preference in the implicit feedback based system, including pointwise preference on an item, and pairwise preferences over two items.

The assumption of pointwise preference [20] is denoted as,

$$Pref(u,i) = 1, Pref(u,j) = 0, i \in I_t^u, j \in I_{nt}^u;$$

where $Pref(u,i) = 1$ denotes user $u$'s positive preference towards item $i$ that she performed *target action* on, while $Pref(u,j) = 0$ represents that user $u$ shows no preference on item $j$ that she did not perform *target action* on. On the contrary, the assumption of pairwise preferences [23] focuses on the relative preferences between the two types of items, which can be denoted as:

$$Pref(u,i) \succ Pref(u,j), i \in I_t^u, j \in I_{nt}^u,$$

This assumption suggests that user $u$ is more likely to prefer item $i$ with *target action*, to item $j$ without *target action*. It can be further relaxed to a user's preference on a set of items [30] instead of a single item, defined as,

$$Pref(u,\mathcal{I}) \succ Pref(u,\mathcal{J}), \mathcal{I} \subseteq I_t^u, \mathcal{J} \subseteq I_{nt}^u,$$

where $Pref(u,\mathcal{I}) = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} Pref(u,i)$ is user $u$'s averaged preference on all items in item-set $\mathcal{I}$; $|\mathcal{I}|$ is the size of item-set $\mathcal{I}$. This relaxed assumption provides us with more accurate pairwise preference relationships [30].

The above assumptions are merely suitable for homogeneous implicit feedback system, since items are classified into two types, i.e., with *target action* and without *target action*. Users prefer items with *target action* to those without *target action*. While in a heterogeneous system, as both *target action* and *auxiliary action* are taken into account, the item classification can be more fine-grained. Items without *target action* but *auxiliary action* are more likely to obtain *target action* than those without any actions. For example, users often browse the details of items that they interested in besides purchasing them. Hence, we assume that user preference towards items with only *auxiliary action* is weaker than those with *target action*, but stronger than those with no action. In other words, *target action*, *auxiliary action* and no action indicate strong, medium and weak user preference, respectively. Note that unlike [18], in our *series pairwise assumption*, the user preference level indicated by different *auxiliary action* are not restricted, since we find that the preference level might be counterintuitive in some scenarios, which will be discussed later.

Inspired by [30], we make our assumption in terms of item-sets rather than a single item. The item-set group for each user is given by,

$$G(u) = \{(\mathcal{I}, \mathcal{J}, \mathcal{K}) \mid \mathcal{I} \subseteq I_t^u, \mathcal{J} \subseteq I_{oa}^u, \mathcal{K} \subseteq I_n^u\}.$$

For any user $u$, her preference towards items in the item-set group, i.e.,$(\mathcal{I}, \mathcal{J}, \mathcal{K})$, is in a descending order,

$$Pref(u, \mathcal{I}) \succ Pref(u, \mathcal{J}) \succ Pref(u, \mathcal{K}). \tag{1}$$

Equation 1 is the user preference assumption for heterogeneous implicit feedback systems, and we name it *series pairwise assumption*. This assumption can be easily extended if more fine-grained item classification is considered. One example is the *negative action*, i.e., *delete* in the Xing dataset, where users can delete items that they do want to see anymore. It shows an obvious negative attitude of the users towards the deleted items. In this scenario, the item-set group for each user $G(u)$ can be reformulated as,

$$G'(u) = \{(\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{G}) \mid \mathcal{I} \subseteq I_t^u, \mathcal{J} \subseteq I_{oa}^u, \mathcal{K} \subseteq I_n^u, \mathcal{G} \subseteq I_d^u\}.$$

Thus, user $u$'s preference over items in the item-set group, i.e., $(\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{G})$ is defined as,

$$Pref(u, \mathcal{I}) \succ Pref(u, \mathcal{J}) \succ Pref(u, \mathcal{K}) \succ Pref(u, \mathcal{G}). \tag{2}$$

In the above assumption, we contend that user's preference towards items with no action is stronger than those with *delete*, as no action implies no user preference, whereas *delete* indicates an obvious negative attitude.

Till now, we have properly defined our *series pairwise assumption* for heterogeneous implicit feedback systems. Next we will elaborate how to seamlessly incorporate this assumption into the basic BPR model for better recommendation.

### *3.2. The Model*

Assuming that all users' preferences are independent from each other [23], the likelihood of preference for all users can be expressed as follows:

$$\begin{aligned}
&\prod_{u\in U}\prod_{(\mathcal{I},\mathcal{J},\mathcal{K})\subseteq I} P\left(\hat{r}_{u\mathcal{I}} > \hat{r}_{u\mathcal{J}} > \hat{r}_{u\mathcal{K}}\right)^{\delta((\mathcal{I},\mathcal{J},\mathcal{K})\in G(u))} \\
&\big(1 - P\left(\hat{r}_{u\mathcal{I}} > \hat{r}_{u\mathcal{J}} > \hat{r}_{u\mathcal{K}}\right)\big)^{(1-\delta((\mathcal{I},\mathcal{J},\mathcal{K})\in G(u)))} \\
=&\prod_{u\in U}\prod_{(\mathcal{I},\mathcal{J},\mathcal{K})\in G(u)} P\left(\hat{r}_{u\mathcal{I}} > \hat{r}_{u\mathcal{J}}\right) P\left(\hat{r}_{u\mathcal{J}} > \hat{r}_{u\mathcal{K}}\right) \\
&\big(1 - P\left(\hat{r}_{u\mathcal{I}} < \hat{r}_{u\mathcal{J}}\right)\big)\big(1 - P\left(\hat{r}_{u\mathcal{J}} < \hat{r}_{u\mathcal{K}}\right)\big)
\end{aligned} \tag{3}$$

where $\hat{r}_{u\mathcal{I}} = |\mathcal{I}|^{-1}\sum_{i\in\mathcal{I}} r_{ui}$ is the estimated averaged preference of user $u$ on items in the item-set $\mathcal{I}$ [30]; $\hat{r}_{ui}$ is the estimated preference of user $u$ towards item $i$; $\delta$ is an indicator function with $\delta(x) = 1$ if $x$ is true, otherwise $\delta(x) = 0$.

Following [23], we utilize sigmoid function $\sigma(x - y) = 1/(1 + e^{-(x-y)})$ to approximate the probability $P(x > y)$, and adopt log-likelihood to reduce the computational complexity. Then, we reach the objective function for our BPRH model,

$$\max f(\theta) = \sum_{u}\left\{\sum_{\mathcal{I},\mathcal{J}} \ln\sigma\left(\frac{\hat{x}_{u\mathcal{I}\mathcal{J}}(\theta)}{\alpha_u}\right) + \sum_{\mathcal{J},\mathcal{K}} \ln\sigma\left(\hat{x}_{u\mathcal{J}\mathcal{K}}(\theta)\right)\right\} - \mathcal{R}(\theta), \tag{4}$$

where $\theta = \{\hat{U}_u \in \mathbb{R}^{|U|\times f}, \hat{V}_i \in \mathbb{R}^{|I|\times f}, b_i \in \mathbb{R}, u \in U, i \in I\}$ is a set of model parameters to be learnt; $\hat{U}_u$ is the user-specific latent feature vector of user $u$, $\hat{V}_i$ is the item-specific latent feature vector of item $i$; $|U|, |I|$ are the numbers of users and items, respectively; $f$ is the dimension of latent factors; $b_i$ is the item bias; $\hat{x}_{u\mathcal{I}\mathcal{J}}(\theta) = \hat{r}_{u\mathcal{I}}(\theta) - \hat{r}_{u\mathcal{J}}(\theta)$ is the estimated preference difference of $u$ for $\mathcal{I}$ with *target action* and $\mathcal{J}$ with *auxiliary action*; $\mathcal{R}(\theta) = \sum_u \sum_{\mathcal{I},\mathcal{J},\mathcal{K}}\{\lambda_u \|U_u\|^2 +$

$\lambda_v(\|V_{\mathcal{I}}\|^2 + \|V_{\mathcal{J}}\|^2 + \|V_{\mathcal{K}}\|^2) + \lambda_b(b_{\mathcal{I}}^2 + b_{\mathcal{J}}^2 + b_{\mathcal{K}}^2)\}$ is the regularization term to avoid overfitting.

Note that there is another important parameter $\alpha_u$ in the objective function $f(\theta)$ controlling the contribution of $\hat{x}_{u\mathcal{I}\mathcal{J}}(\theta)$ in Equation 4. As mentioned in Section 1, *auxiliary action* is observed to have correlation with *target action* since they often co-occur. This correlation, indicating a user's pattern of performing actions, can be significantly different among users. We call it the *auxiliary-target correlation*, denoted by $C^u$. For those users whose $C^u$ is high, a *target action* will have a high probability to occur after users performed *auxiliary action* on items. In this case, the item with *auxiliary action* obtains relatively strong user preference, thus $P(\hat{r}_{u\mathcal{I}} > \hat{r}_{u\mathcal{J}})$ should be relatively small, vice versa. Hence, the penalty weight $\alpha_u$ is introduced in Equation 4 to better handle this issue. $\alpha_u$ is a user-specific function and positively correlated with $C^u$. Specifically, for any user $u$, if the value of $C^u$ is high, and $\alpha_u$ is large, thus the contribution of $\hat{x}_{u\mathcal{I}\mathcal{J}}(\theta)$ is small, which is equivalent to a small value of $P(\hat{r}_{u\mathcal{I}} > \hat{r}_{u\mathcal{J}})$. And similar properties hold if the value of $C^u$ is low. In a word, $\alpha_u$ greatly facilitates to better embed our *series pairwise assumption*, and we will discuss how to calculate the value of this important parameter in the next subsection.

The effect of parameters in the sigmoid function was investigated by Mao et al. [35]. They found that in the objective function of a standard BPR model, parameter like $\alpha_u$ in the sigmoid would affect the gradients of $\theta$ and sometimes boost the popularity tendency, i.e., the tendency of an algorithm to recommend the popular items. Unlike the BPR which contains one sigmoid term, Equation 4 contain two sigmoid terms, where one term is with parameter. In these cases, the parameter simply adjusts the gradient of one term, while that of the other term remains impervious. Therefore, the popularity tendency of BPRH can not by scaled up by $\alpha_u$. This conclusion is further verified by empirical studied in Section 5.

The objective function shown in Equation 4 is based on the *series pairwise assumption* in Equation 1. For the scenario where *negative action*, i.e., *delete*, is available (see Equation 2), Equation 4 can be reformulated as follows,

$$\begin{aligned}\max f'(\theta) = \sum_{u} \Bigg\{ & \sum_{\mathcal{I},\mathcal{J}} \ln \sigma \left( \frac{\hat{x}_{u\mathcal{I}\mathcal{J}}(\theta)}{\alpha_u} \right) + \sum_{\mathcal{J},\mathcal{K}} \ln \sigma \left( \hat{x}_{u\mathcal{J}\mathcal{K}}(\theta) \right) \\ & + \sum_{\mathcal{K},\mathcal{Z}} \ln \sigma \left( \hat{x}_{u\mathcal{K}\mathcal{Z}}(\theta) \right) \Bigg\} - \mathcal{R}(\theta) ,\end{aligned} \tag{5}$$

### *3.3. Auxiliary-Target Correlation based on Co-occurrence*

As illustrated in the last subsection, parameter $\alpha_u$ is positively influenced by the *auxiliary-target correlation* of user $u$, i.e., $C^u$. We, therefore, calculate the value of $\alpha_u$ by quantifying $C^u$ for each $u$.

In heterogeneous implicit feedback systems, a user can perform multiple actions on a same item. That is to say, *auxiliary action* often co-occur with *target action*, which has been verified by the data statistics in Table 1 (Section 1) and Figure 2 (Section 2). We contend that the co-occurrence of *auxiliary action* and *target action* indicates the correlation between them, i.e., $C^u$. And the higher the frequency of co-occurrences, the larger the value of $C^u$ is, vice versa. For instance, in an online shopping system, Alice would like to *view* a number of items just for fun, no matter she is willing to *purchase* one or not, while Bob only check the details of those items he has bought or would like to buy at the moment. According to different user patterns of performing action, We draw a conclusion that the value of $C^u$ of Bob is higher than that of Alice, since an item viewed by Bob is more likely to be purchased (or have been purchased), i.e., the co-occurrence frequency of *auxiliary action* and *target action* for Bob is higher than that of Alice.

Thus, we first quantify $C^u$ via the co-occurrence frequency of *auxiliary action* and *target action* for user $u$, which can be calculated by ratio of co-occurrence. However, we find that the ratio of co-occurrence between *auxiliary action* and *target action*, denoted by $C^u_{at}$, and that of co-occurrence between *target action* and *auxiliary action*, denoted by $C^u_{ta}$, are asymmetric, shown as follows,

$$C^u_{at} = \frac{|I^u_a \cap I^u_t|}{|I^u_t|}, \quad C^u_{ta} = \frac{|I^u_a \cap I^u_t|}{|I^u_a|},$$

where $|I|$ is the size of item set $I$; $C^u_{at}$ denotes the relative amount of *auxiliary action* which co-occur with each *target action*; while $C^u_{ta}$ denotes the relative amount of *target action* which co-occur with each *auxiliary action*. Both of them would influence the value of $C^u$. Hence, we define $C^u$ by the combination influence of $C^u_{at}$ and $C^u_{ta}$, given by,

$$C^u(X) = \frac{2 \cdot C^u_{ta}(X) \cdot C^u_{at}(X)}{C^u_{ta}(X) + C^u_{at}(X)} \tag{6}$$

where $X \in \{\textit{view, like}\}$ represents the types of *auxiliary action*. Since $\alpha_u$ has positive relationships with $C^u$, it can be formulated as follows,

$$\alpha_u(X) = \omega \cdot C^u(X), \tag{7}$$

where $\omega > 0$ is the coefficient controlling the importance of $C^u(X)$. $\omega$ is a hyperparameter measuring the value of $\alpha_u(X)$ when $C^u(X) = 1$.

In general, for user $u$ who performs $m$ kinds of *auxiliary action*, $X_1, X_2, \cdots, X_m$, the final $\alpha_u$ is given by:

$$\alpha_u = \sum_{r=1}^{m} \rho_r \cdot \alpha_u(X_r), \tag{8}$$

where $\sum_{r=1}^{m} \rho_r = 1, \rho_r > 0$ controls the importance of *auxiliary action* $X_r$. Note that in Equation 4 and 5, if $\mathcal{J} \subseteq I^u_{oa}$ does not contain items with *auxiliary action* $X_r$ from $u$, we set $\alpha_u(X_r) = 0$ when calculating $\alpha_u$. Specifically in our study, two types of *auxiliary action* (i.e., *view* and *like*) are taken into consideration. Hence, we can compute $\alpha_u(\textit{view})$ and $\alpha_u(\textit{like})$, respectively. And Equation 8 is rewritten as,

$$\alpha_u = \rho \cdot \alpha_u(\textit{view}) + (1 - \rho) \cdot \alpha_u(\textit{like}) \tag{9}$$

where $\rho \in [0, 1]$. If $u$ only performs *view*/*like*, then $\rho = 1/0$.

### *3.4. 'Like' in Different Scenarios*

With the definition of *auxiliary-target correlation*, we calculate the overall correlation w.r.t. *view*, i.e., $C(\textit{view})$, and *like*, i.e., $C(\textit{like})$ for all users in the real-world datasets, according to the following equation:

$$C(X) = \frac{2 \cdot C_{ta}(X) \cdot C_{at}(X)}{C_{ta}(X) + C_{at}(X)}, \tag{10}$$

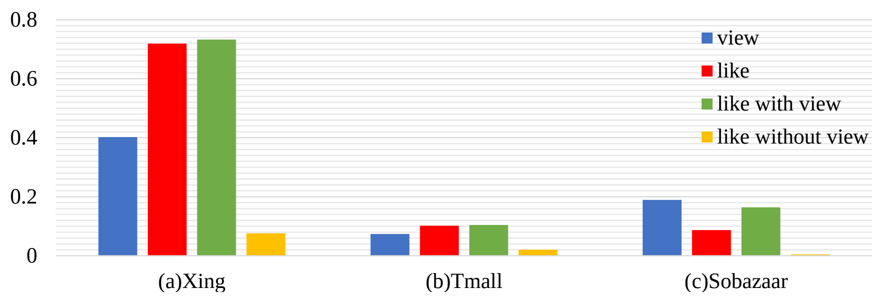



Figure 3: The overall *auxiliary-target Correlation* of three datasets representing different scenarios.

where $C_{ta}(X) = |I_x \cap I_t|/|I_t|, C_{at}(X) = |I_x \cap I_t|/|I_x|$; $X \in \{view, like\}$, $I_x \in \{I_v, I_l\}$. The results on both datasets are shown in Figure 3. Note that the major difference between the three datasets is that there is a social network built in Sobazaar, whereas Xing and Tmall do not have social networks.

Intuitively, *like*, as a straightforward way to express user interests on items, should correlate more with *target action* than *view*, i.e., $C(like) > C(view)$. This can be verified by the results on Xing, shown in Figure 3 (a), where $C(view) = 0.4023$ and $C(like) = 0.7190$, as well as the results on Tmall, shown in Figure 3 (b). However, an opposite observation is found in Figure 3 (c) showing the results on Sobazaar, where $C(view) = 0.1892$, whereas $C(like) = 0.0871$. This counterintuitive observation is probably caused by the different types of services offered by the three online systems. Sobazaar provides a social network where users can follow each other and catch up with the latest happenings of their friends. With the social network, user can not only '*like*' items they are interested in but also those items their friends are fond of. For instance, people share something cool on their homepages and their friends would click '*like*' as support. In other words, in the systems with a social network, e.g., Sobazaar, due to the complicated motivation for a user to click '*like*', some *like* may have little to do with user preference.

Therefore, to filter out the noise in *like*, we select items with both *like* and *view* to compose the *filtered like* item set, shown as follows,

$$I_{fl} = \{i | i \in I_v \cap I_l\}, \tag{11}$$

since most users would view an item if they are truly into it. In Figure 3 (c), we can see that the *filtered like* correlates much more to *target action* than the original *like*, whereas the *auxiliary-target correlation* of *like* without *view* is close to zero. The same filtering process has been done on Xing and Tmall to decrease the noise in *like*. The results are depicted by Figure 3(a),(b), where we can observe that the improvements on *auxiliary-target correlation* are not significant. Hence, we make our conclusion that filtering out noise in *like* in the scenario where social networks are involved can improve the *auxiliary-target correlation*, thus to enhance recommendation performance.

In the scenario with social network, another possible concern is that whether user interaction affects the independence of users' preferences. In fact, social network is often very sparse. Its influence on user preference is generally ignored by many studies, including all related works mentioned in our paper. In our model, this influence is further reduced by action filtering. We do not consider social influence on users' preferences and hold the assumption accordingly.

In practical application, whether to enable the filtering process could be determined according to background information of an authorized dataset. For anonymous datasets, a simple comparison of the overall *auxiliary-target correlation* of *filtered like* and *like* could be done before applying the proposed algorithm. If the former is lager and gap is big (e.g., $> 20\%$), the filtering process will be enabled.

## 4. Optimization for BPRH

We adopt the widely used stochastic gradient descent (SGD) algorithm [36, 37] to optimize the objective function in Equation 4 and 5. For each user $u$, items are sampled according to the item-set group $G(u)$. We then calculate the derivative of model parameters and update $\theta$ by going along the ascending gradient direction with learning rate $\gamma$ [30, 23],

$$\theta \leftarrow \theta + \gamma \Delta\theta, \tag{12}$$

where $\Delta\theta$ is the derivative set of model parameters, shown as follows [30, 23],

$$\Delta\theta = \begin{cases} \Delta U = \dfrac{\partial f(\theta)}{\partial U} - \lambda_u U \\ \Delta V = \dfrac{\partial f(\theta, \mathcal{I})}{\partial V_i} + \dfrac{\partial f(\theta, \mathcal{J})}{\partial V_j} + \dfrac{\partial f(\theta, \mathcal{K})}{\partial V_k} + \dfrac{\partial f(\theta, \mathcal{G})}{\partial V_g} \\ \qquad - \lambda_v \left(V_{\mathcal{I}} + V_{\mathcal{J}} + V_{\mathcal{K}} + V_{\mathcal{G}}\right) \\ \Delta B = \dfrac{\partial f(\theta, \mathcal{I})}{\partial B_i} + \dfrac{\partial f(\theta, \mathcal{J})}{\partial B_j} + \dfrac{\partial f(\theta, \mathcal{K})}{\partial B_k} + \dfrac{\partial f(\theta, \mathcal{G})}{\partial B_g} \\ \qquad - \lambda_b \left(B_{\mathcal{I}} + B_{\mathcal{J}} + B_{\mathcal{K}} + B_{\mathcal{G}}\right) \end{cases}, \tag{13}$$

where $\frac{\partial f(\theta,\mathcal{I})}{\partial V_i} = \sum_{i\in\mathcal{I}} |\mathcal{I}|^{-1} \frac{\partial f(\theta)}{\partial V_i}$, $V_{\mathcal{I}} = \sum_{i\in\mathcal{I}} |\mathcal{I}|^{-1} V_i$ and $\frac{\partial f(\theta,\mathcal{I})}{\partial B_i} = \sum_{i\in\mathcal{I}} |\mathcal{I}|^{-1} \frac{\partial f(\theta)}{\partial B_i}$, $B_{\mathcal{I}} = \sum_{i\in\mathcal{I}} |\mathcal{I}|^{-1} B_i$. If *negative action* (e.g., *delete*) is not available, we set $\frac{\partial f(\theta,\mathcal{G})}{\partial V_g} = \frac{\partial f(\theta,\mathcal{G})}{\partial B_g} = V_{\mathcal{G}} = B_{\mathcal{G}} = 0$. Note that when $I_v \cup I_l = \emptyset$, which means there is no *auxiliary action* performed by users, BPRH is then downgraded to the basic BPR model [23].

**Item-set based on *co-selection*.** In our BPRH model, we study user preference over an item-set instead of a single item, hence, the first key part in our model learning process is to build item-set for each user. Since randomly picking items to build the item-sets suffers from inconsistent user preference especially when the size of item-sets is large [30], we generate item-sets by selecting items that share similar features. A simple but effective scheme is based on *co-selection*, namely selecting items with *target action* that are performed by at least two users at the same time. For any item $i$, $S^i$ is the set of items that share the feature, i.e., *co-selection*, with item $i$, shown as,

$$S^i = \{j \mid |U^i \cap U^j| \geq 2, i, j \in \mathcal{I}\},$$

where $U^i, U^j$ are the sets of users who perform *target action* on item $i$ and $j$, respectively; Items in $S^i$ that share *co-selection* feature, are more likely to obtain similar user preference than randomly picked items. In our model, we set $\mathcal{I}, \mathcal{J}, \mathcal{K} \subseteq S^i$, if $i \in \mathcal{I}, \mathcal{J}, \mathcal{K}$.

Now we present an example to elaborate the process of building the item-set $\mathcal{I}$ for a given user. Table 3 shows the synthetic purchase record of several users. Suppose we intend to build an item-set $\mathcal{I}$ for $u_2$. First, an item $i \in I_t^{u_2} = \{i_1, i_2, i_6\}$ is sampled, say $i_2$. Then we set $\mathcal{I} = I_t^{u_2} \cap S^{i_2}$, where $S^{i_2} = \{i_2, i_4, i_6, i_8\}$. Finally, $\mathcal{I} = \{i_2, i_6\}$.

Table 3: The synthetic data of users' purchase record, where '1' means the user purchased the item.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ | $i_9$ |
|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | 1 | | 1 | | 1 | | | | 1 |
| $u_2$ | 1 | 1 | | | | 1 | | | |
| $u_3$ | | | 1 | | | 1 | | | |
| $u_4$ | | 1 | | 1 | | 1 | | 1 | |
| $u_5$ | 1 | | | | 1 | | | | |
| $u_6$ | | 1 | | 1 | | | 1 | 1 | 1 |
| $u_7$ | | 1 | | | | | 1 | 1 | |
| $u_8$ | 1 | | | | 1 | | | | |
| $u_9$ | | | | 1 | | | | 1 | |

**The adaptive sampling strategy.** Another essential part in the model learning process is the sampling strategy. For the scenario where only one type of *auxiliary action* (e.g., *view* or *like*) is available, we uniformly sample $i \in I_t^u, j \in I_{oa}^u, k \in I_n^u, g \in I_d^u$ to build the item-sets $\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{G}$ for user $u$. However, when multiple types of *auxiliary action* (e.g., *view* and *like*) are taken into account, the *auxiliary-target correlation* can be significantly different among different types of *auxiliary action*. In this case, uniform sampling would not be suitable anymore. For instance, in our study, if $C^u(like) < C^u(view)$, i.e., *like* correlates less with *target action* than *view*, items in $I_l^u$ should be less frequently sampled, since they are less correlated with *target action*, thus to make less contributions to improve the recommendation performance, vice versa. To address this issue, we propose an adaptive sampling strategy based on *auxiliary-target correlation*. Items that belong to $I^u_{oa}$ are drawn according to a *Bernoulli distribution* $p(\mu)$,

$$p(\mu) := \epsilon^{\mu}(1-\epsilon)^{1-\mu}, \mu = 0, 1. \tag{14}$$

We pick an item from $I^u_{oa} \cap I^u_l$ if $\mu = 1$ and from $I^u_{oa} \cap I^u_v$ otherwise. The probability $\epsilon$ that determines the sample frequency of items in $I^u_{oa} \cap I^u_l$ is given by,

$$\epsilon = \frac{1}{1 + e^{-10\cdot(C^u(like) - C^u(view))}}, \tag{15}$$

and the sample frequency of items in $I^u_{oa} \cap I^u_v$ would be $(1-\epsilon)$. Equation 15 can help reduce the negative impact of items in $I^u_{oa}$ that are less correlated with the *target action*. Specifically, the more $C^u(like)$ is larger than $C^u(view)$, the larger $\epsilon$ should be, hence, the more frequently items in $I^u_l$ is sampled.

The detailed learning algorithm of BPRH is presented in Algorithm 1. To explain, several arguments are taken as input, including the datasets $T(u,i)$, $A(u,i)$, $N(u,i)$; parameters $\omega$, $\rho$; the regularization parameters $\lambda_u$, $\lambda_v$, $\lambda_b$; and the initial learning rate $\gamma$. First, we calculate the *auxiliary-target correlation* $C$ for users and generate $S$ for items (line 1-2). Then we iterate the training process until the objective function converges (line 3). Specifically, we uniformly draw users (line 4) and sample items by using the adaptive strategy to build item-sets (line 5-24). The model parameters $\theta$ are updated during the training process to approach the optimal solution (line 25).

The time complexity of the learning algorithm for BPRH is $O(\max(|\mathcal{I}|, |\mathcal{J}|, |\mathcal{K}|)$ $|U|f{*}\#Iter)$, where $|\mathcal{I}|, |\mathcal{J}|, |\mathcal{K}|$ are sizes of item-sets $\mathcal{I}, \mathcal{J}, \mathcal{K}$, respectively; $|U|$ is the number of users; $f$ is the dimension of latent factors; $\#Iter$ is the number of iterations. In practice, the values of $|\mathcal{I}|, |\mathcal{J}|, |\mathcal{K}|$ are usually small (less than 5), since the real-world datasets are very sparse. Therefore, the efficiency of our proposed method is comparable to BPR whose time complexity is $O(|U|f * \#Iter)$,

**Algorithm 1:** The learning algorithm for BPRH

**Input:** $T(u,i), A(u,i), N(u,i), \omega, \rho, \lambda_u, \lambda_v, \lambda_b, \gamma$.

**Output:** $\theta = \{\hat{U}_u, \hat{V}_i, b_i\}$

Calculate *auxiliary-target correlation* $C$ for every user and each types of *auxiliary action*.

Generate $S$ for each item.

**for** *iterations* **do**

    Uniformly sample a user $u \in U$;

    Uniformly sample a item $i \in I_t^u$;

    Build $\mathcal{I} = I_t^u \cap S^i$;

    **if** $I_{oa}^u \neq \emptyset$ **then**

        **if** $I_l^u \neq \emptyset$ *and* $I_v^u \neq \emptyset$ **then**

            Calculate the $\epsilon$ according to Equation 15;

            Generate $\mu$ according to *Bernoulli distribution* $p(\mu)$;

            **if** $\mu = 1$ **then**

                Pick an item $j \in I_{oa}^u \cap I_l^u$ randomly;

                Build $\mathcal{J} = I_{oa}^u \cap I_l^u \cap S^i$;

            **else**

                Pick an item $j \in I_{oa}^u \cap I_v^u$ randomly;

                Build $\mathcal{J} = I_{oa}^u \cap I_v^u \cap S^i$;

        **else**

            Pick an item $j \in I_{oa}^u$ randomly;

            Build $\mathcal{J} = I_{oa}^u \cap S^i$;

    Uniformly sample a item $k \in I_n^u$;

    Build $\mathcal{K} = I_n^u \cap S^i$;

    **if** $I_d^u \neq \emptyset$ **then**

        Pick an item $g \in I_d^u$ randomly;

        Build $\mathcal{G} = I_d^u \cap S^i$;

    Update $\theta$ via Equation 12 and 13.

which is superior to other BPR's variants [22, 30, 26]. The running time[3] of each iteration is 0.5s, 3.4s and 17.1s for Sobazaar, Xing and Tmall respectively.

## 5. Experiments and Analysis

To evaluate the effectiveness of our BPRH model, we conduct comprehensive experiments on three real-world datasets, whereby we want to find out answers to the following two research questions: (1) what is the effect of *auxiliary action* on improving the performance of recommendation; (2) how our proposed method performs in comparison with other state-of-the-art counterparts.

### 5.1. Datasets

Three real-world datasets are utilized in our study, including Sobazaar, Tmall and Xing. The statistics of all the datasets are summarized in Table 4.

Table 4: Statistics of the three real-world datasets.

| **Datasets** | **Sobazaar** | **Xing** | **Tmall** |
|---|---|---|---|
| #user | 4,712 | 107,463 | 424,170 |
| #item | 7,015 | 190,099 | 372,740 |
| #*purchase*/*reply* | 15,208 | 32,7234 | 3,292,144 |
| #*view* | 126,846 | 1,291,305 | 48,550,713 |
| #*like* | 95,589 | 138,168 | 3,305,723 |
| #*delete* | - | 252,964 | - |

Sobazaar[4] is an online fashion portal where users can interact with the portal via an app [38]. Similar as most e-commerce platforms, users can perform *purchase*, *view* and *like* on items. Besides, a social network is also part of

[3] Our experiments run on a PC with Intel Core processor with a speed of 2.2GHz and 8GB of RAM.

[4] shop.villoid.com. The dataset has been released at github.com/hainguyen-telenor/Learning-to-rank-from-implicit-feedback.

the service offered by Sobazaar, where users could become friends with each other. Tmall[5] is a similar but much larger dataset in which *purchase*, *view* and *like* are observed as well. It is generated from a famous online retail website without a social network. Xing[6] is a job hunting website where users can acquire recruitment information posted by companies that are looking for candidates for their jobs. Users can *click* and *bookmark* a job posting or *reply* to it if they intend to apply for the job. They can also *delete* unwanted postings. In this dataset, *reply* is the *target action*; *delete* is the *negative action*; *click* and *bookmark* are *auxiliary action*, which are equivalent to *view* and *like*, respectively. For simplicity, we name them as *view* and *like*.

To the best of our knowledge, Sobazaar, Tmall and Xing are the only publicly available datasets that contain heterogeneous implicit feedback information.

### *5.2. Evaluation Metrics*

We adopt 5-fold cross validation to evaluate the performance of each method. Specifically, we randomly split the *target action* dataset $T(u, i)$ into five folds. In each iteration, four folds are used as the training data (denoted as $T_{tr}$) and the remaining fold as the test data (denoted as $T_{te}$). The *auxiliary action* $A(u, i)$ and *negative action* $N(u, i)$ datasets, if available, would be treated as supplement data to work with the training data during the model learning process. We repeat this procedure five times until all folds are tested and the average results are reported as the final performance. Finally, based on the estimated user preference, a personalized recommendation list will be generated for user $u$, i.e., $I_r^u = \{i_1, i_2, ..., i_m, ...\}$. Six widely used metrics are adopted to evaluate each method, including precision, recall, area under the ROC curve (AUC) [36], normalized discounted cumulative gain (NDCG) [39], mean average precision (MAP) [40] and mean reciprocal rank (MRR) [41].

In terms of precision and recall, if the size of the recommendation list is $K$,

[5] www.tmall.com. The dataset is obtained from tianchi.shuju.aliyun.com.

[6] www.xing.com. The dataset is used for the ACM RecSys Challenge 2016.

i.e., $|I_r^u| = K$, then *precision@K* and *recall@K* are defined as follows,

$$precision@K = \frac{1}{|U|}\sum_{u\in U}\frac{1}{K}\sum_{m=1}^{K}\delta\Big((u,i_m)\in T_{te}\Big),$$

$$recall@K = \frac{1}{|U|}\sum_{u\in U}\frac{1}{|I_{T_{te}}^u|}\sum_{m=1}^{K}\delta\Big((u,i_m)\in T_{te}\Big),$$

where $I_{T_{te}}^u$ is the set of items that user $u$ performs *target action* on, in the test data $T_{te}$. $\delta$ is the indicator function with $\delta(x) = 1$ if $x$ is true, otherwise $\delta(x) = 0$. We set $K = \{5, 10\}$ in this study for both precision and recall.

AUC is defined as follows,

$$AUC = \frac{1}{|U|}\sum_{u\in U}\frac{1}{|E(u)|}\sum_{(i,j)\in E(u)}\delta(\hat{r}_{ui} > \hat{r}_{uj}),$$

where $E(u) = \big\{(i,j)|i \in I_{T_{te}}^u, j \notin I_{T_{te}}^u\big\}$.

NDCG is given by,

$$NDCG@K = \frac{1}{|U|}\sum_{u\in U}\frac{1}{IDCG_u}\sum_{i=1}^{K}\frac{2^{\delta(i\in I_{T_{te}}^u)}-1}{\log_2(i+1)},$$

where $IDCG_u$ is the normalization term, i.e., the ideal $DCG$ value for $u$ when $|I_r^u| = K$.

MAP is defined as,

$$MAP = \frac{1}{|U|}\sum_{u\in U}\sum_{k=1}^{|I_r^u|}\frac{1}{k}\sum_{m=1}^{k}\delta(i_m \in I_{T_{te}}^u).$$

And MRR is defined as,

$$MRR = \frac{1}{|U|}\sum_{u\in U}\frac{1}{\min_{i_m\in I_{T_{te}}^u} m}.$$

Larger values of these metrics mean better recommendation performance.

### *5.3. Comparison Methods*

We compare BPRH with eight state-of-the-art algorithms.[7]

[7]Source code is included in the LibRec library at www.librec.net.

- **Random:** Randomly select items to users for recommendation;
- **PopRank:** Popularity-based ranking is a simple but effective method which recommends the most popular items in terms of *target actions* to users.
- **BPR:** Bayesian personalized ranking [23] is the first pairwise method for item recommendation; Note that two version of BPR are compared: $\text{BPR}_t$ only considers $T(u, i)$, i.e., the data of *target action* and $\text{BPR}_{ta}$ considers $T(u, i) \cup A(u, i)$, i.e., the data of both *target action* and *auxiliary action*.
- **CoFiSet:** Collaborative filtering via learning pairwise preferences over item-sets [30] generalizes the pairwise BPR assumption by using the relationship between two item-sets instead of two items.
- **BPR++(N):** An extension of BPR that focuses on the pairwise preference towards two items both with actions. The method assumes that items obtained more actions are more preferable [21][8].
- **MF-BPR:** Multi-feedback Bayesian Personalized Ranking [18] is a pairwise method that exploits heterogeneous implicit feedback with an extended sampling method.
- **ABPR:** Adaptive BPR method [26] is the first work that is able to incorporate two types of implicit feedback, i.e., *purchase* and *view*;
- **TJSL:** A novel transfer learning method that recommends items via a joint similarity learning model based on knowledge transfer from *view* to *purchase* [25];

Besides, several variants of our proposed framework are compared, shown as follows,

[8] BPR++(T) as another variant of BPR++ presented in [21] is not included in our discussion since it depends on temporal information, which is beyond the scope of this paper.

- **BPRH**$_{cv}$: only considers *view* and sets $\alpha_u$ to be the same constant for all users;
- **BPRH**$_{v}$: only considers *view*;
- **BPRH**$_{l}$: only considers *like*;
- **BPRH**$_{fl}$: only considers *filtered like*;
- **BPRH**$_{vl}$: considers *view* and *like* for Xing and Tmall; *view* and *filtered like* for Sobazaar. Since no social network is involved in Xing or Tmall, we just adopt the original *like*;
- **BPRH**$'_{vl}$: is a variant of BPRH$_{vl}$, where we randomly select items to build item-sets; Similar settings as in CoFiSet are applied to optimize the size of item-sets;
- **BPRH**$''_{vl}$: is a variant of BPRH$'_{vl}$, where the size of item-sets is fixed to be one. That is to say, the *series pairwise assumption* is downgraded to study pairwise user preference over a single item;
- **BPRH**$_{vld}$: considers *view*, *like* and *delete*, which only works for Xing, since Sobazaar and Tmall do not contain *delete* action.

We empirically find out the optimal parameter settings for all the methods. Specifically, we apply a grid search in $\{0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0\}$ for the regularization parameters $\lambda_u, \lambda_v, \lambda_b$, and a grid search in $\{0.0001, 0.001, 0.01, 0.1\}$ for the learning rate $\gamma$; The dimension of latent factors is set to $f \in \{10, 20, 50, 100, 200\}$; For CoFiSet, we apply a grid search in $\{1, 2, 3, 4, 5\}$ to find out the optimal size of item-sets [30]; For BPR++(N) and MF-BPR, the sampling parameter $\beta$ are searched in $[0, 1]$ with the step of 0.05 [21, 18]. For ABPR and TJSL, the outer iteration number $L = 10$ and the number of models for final prediction $L_0 = 3$ [26, 25]. For ABPR, we search the empirical error threshold $\tau$ in $[100, 1000]$ with the step of 50 [26].

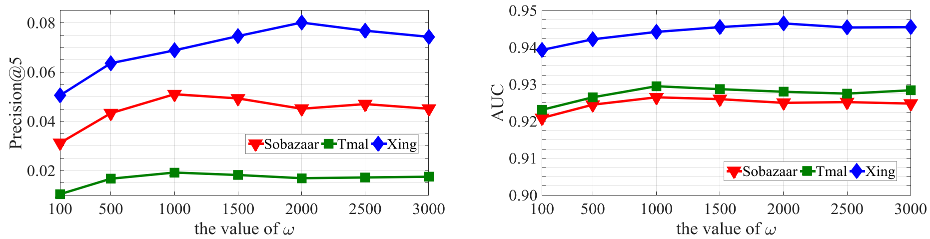
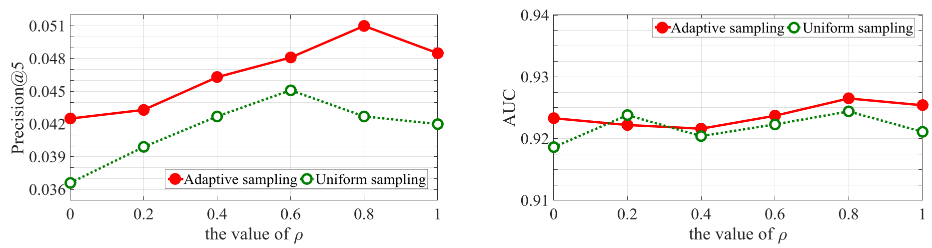
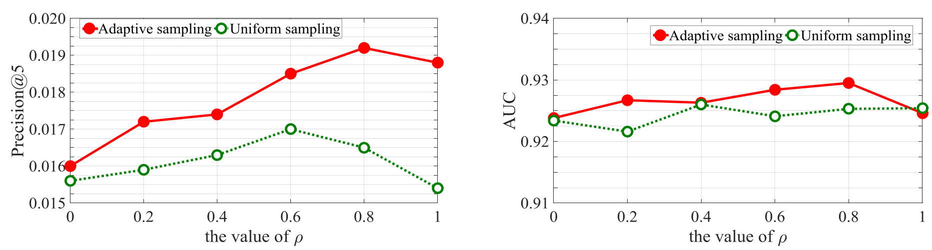


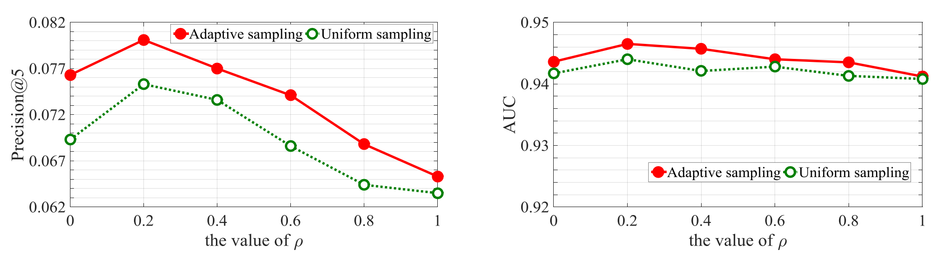

Figure 4: The impacts of $\omega$ on Sobazaar, Tmall and Xing datasets.




(a) Sobazaar




(b) Tmall




(c) Xing

Figure 5: The impacts of $\rho$ and sampling strategies on Sobazaar, Tmall and Xing datasets.

### *5.4. Results and Analysis*

In this subsection, we present the experimental results on the three real-world datasets. Through insightful analysis on the results, the two research questions mentioned above can be well answered.

**The Impact of Parameter $\omega$.** In Equation 7, $\omega$ controls the importance of *auxiliary-target correlation* (i.e., $C^u$). We apply a grid search in $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$ to study its impact on recommendation performance. Figure 4 depicts the results of BPRH$_{vl}$ on Sobazaar, Tmal and Xing. For precision, as $\omega$ increases, it improves at first on all datasets. The best performance on Sobazaar is achieved when $\omega = 1000$ with 63% of improvement relative to the worst results with $\omega = 100$ , while for Tmall and Xing, the peak values are obtained when $\omega = 1000$ and $\omega = 2000$ with the improvement of 85% and 59% relative to the worst results, respectively. As $\omega$ further increases, precision slightly goes down, and finally remains stable on all datasets. For AUC, it keeps relatively stable as $\omega$ changes. Less than 1% of improvements for the best case with $\omega = 1000$ on Sobazaar and Tmall, with $\omega = 2000$ on Xing are achieved relative to the worst results with $\omega = 100$, respectively. In summary, the performance variations across datasets suggest the need of dataset-specific settings for $\omega$; the similarity in performance variation across $\omega$ values demonstrates the robustness of our proposed model. In this study, we set $\omega = 1000$ for Sobazaar and Tmall, $\omega = 2000$ for Xing.

**The Impact of Parameter $\rho$.** In Equation 9, $\rho$ controls the contribution of the two types of *auxiliary action*, i.e., *view* and *like*. We apply a grid search in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ to analyze the impact of $\rho$ with the uniform and adaptive sampling strategies. The results are shown in Figure 5 (a) (b) (c) for Sobazaar, Tmall and Xing, respectively, where several interesting observations are noted: (1) For all datasets, with the varying of $\rho$, the performance of the adaptive sampling generally outperforms that of uniform sampling in terms of both precision and AUC, indicating the effectiveness of our adaptive sampling strategy. (2) The best performances of Sobazaar, Tmall and Xing are all achieved with

$\rho$ around $0.2 \sim 0.8$, suggesting that only appropriate combination of *view* and *like* can help obtain the best results. (3) The peak values of both precision and AUC are obtained at $\rho = 0.8$ for Sobazaar and Tmall, while $\rho = 0.2$ for Xing. This also verifies the fact (shown in Figure 3) that *view* is the *auxiliary action* with larger overall *auxiliary-target correlation* in Sobazaar and Tmall, while *like* is the one with higher *auxiliary-target correlation* in Xing. (4) Compared with precision, AUC is less sensitive to $\rho$. In our study, we set $\rho = 0.8$ for Sobazaar and Tmall, $\rho = 0.2$ for Xing.

Based on the above analysis, the two hyperparameters $\omega$ and $\rho$ are not difficult to tune. A relatively large value ($> 1000$) for $\omega$ and a value between 0.2 and 0.8 for $\rho$ are recommended as initialization. Note that the number of hyperparameters in BPRH is no more than that in most of the baseline methods[9].

**Results of BPRH Variants.** The results of different variants of BPRH are presented in Table 5, and a number of findings are summarized as follows:

- The performance of $\mathrm{BPRH}_v$ is better than that of $\mathrm{BPRH}_{cv}$ on the three datasets. Although both of them consider the same *auxiliary action*, i.e., *view*, $\mathrm{BPRH}_{cv}$ sets all $\alpha_u$ to be the same constant, whereas $\mathrm{BPRH}_v$ treats all users distinctively by calculating $\alpha_u$ for each user. The results imply that the different patterns of performing *auxiliary action* among different users should be fully taken into consideration for better recommendation.
- Although both $\mathrm{BPRH}_l$ and $\mathrm{BPRH}_{fl}$ consider the same *auxiliary action*, i.e., *like*, the latter utilizes the *filtered like* generated via Equation 11. $\mathrm{BPRH}_{fl}$ is found to achieve much better performance than $\mathrm{BPRH}_l$ on Sobazaar. In contrast, no significant improvements even some worse results in a few metrics are observed on Tmall and Xing. These results coincide with the conclusion made in Figure 3: the effect on improving

[9]The number of hyperparameters (except for those share by all) in BPR, CoFiSet, BPR++, MF-BPR, ABPR, TJSL, BPRH is 0, 2, 1, 4, 3, 3, 2, respectively.

Table 5: Recommendation performance of variants of BPRH on the three real-world datasets ($f = 50$).

| **Dataset** | **Method** | **Pre@5** | **Pre@10** | **Rec@5** | **Rec@10** | **AUC** | **MAP** | **NDCG** | **MRR** |
|---|---|---|---|---|---|---|---|---|---|
| Sobazaar | $\text{BPRH}_{cv}$ | 0.0254 | 0.0193 | 0.0821 | 0.1347 | 0.8824 | 0.0721 | 0.1803 | 0.0662 |
| | $\text{BPRH}_{v}$ | 0.0355 | 0.0259 | 0.1452 | 0.2040 | 0.9030 | 0.1059 | 0.2473 | 0.1141 |
| | $\text{BPRH}_{l}$ | 0.0174 | 0.0132 | 0.0642 | 0.0927 | 0.8246 | 0.0471 | 0.1767 | 0.0568 |
| | $\text{BPRH}_{fl}$ | 0.0191 | 0.0144 | 0.0707 | 0.1005 | 0.8449 | 0.0490 | 0.1927 | 0.0611 |
| | $\text{BPRH}''_{vl}$ | 0.0434 | 0.0317 | 0.1728 | 0.2377 | 0.9034 | 0.1205 | 0.2557 | 0.1302 |
| | $\text{BPRH}'_{vl}$ | 0.0472 | 0.0331 | 0.1829 | 0.2526 | 0.9128 | 0.1294 | 0.2623 | 0.1377 |
| | $\text{BPRH}_{vl}$ | 0.0510 | 0.0355 | 0.2005 | 0.2704 | 0.9265 | 0.1378 | 0.2807 | 0.1504 |
| Tmall | $\text{BPRH}_{cv}$ | 0.0086 | 0.0075 | 0.0281 | 0.0522 | 0.8753 | 0.0250 | 0.1501 | 0.0304 |
| | $\text{BPRH}_{v}$ | 0.0093 | 0.0091 | 0.0333 | 0.0631 | 0.9082 | 0.0275 | 0.1645 | 0.0351 |
| | $\text{BPRH}_{l}$ | 0.0166 | 0.0113 | 0.0492 | 0.0669 | 0.6852 | 0.0361 | 0.1399 | 0.0528 |
| | $\text{BPRH}_{fl}$ | 0.0164 | 0.0114 | 0.0492 | 0.0668 | 0.6803 | 0.0360 | 0.1400 | 0.0527 |
| | $\text{BPRH}''_{vl}$ | 0.0173 | 0.0152 | 0.0591 | 0.1001 | 0.9037 | 0.0432 | 0.1710 | 0.0519 |
| | $\text{BPRH}'_{vl}$ | 0.0181 | 0.0158 | 0.0627 | 0.1054 | 0.9124 | 0.0459 | 0.1790 | 0.0553 |
| | $\text{BPRH}_{vl}$ | 0.0192 | 0.0165 | 0.0668 | 0.1121 | 0.9295 | 0.0486 | 0.1904 | 0.0624 |
| Xing | $\text{BPRH}_{cv}$ | 0.0384 | 0.0298 | 0.1956 | 0.2128 | 0.8654 | 0.1586 | 0.2089 | 0.1576 |
| | $\text{BPRH}_{v}$ | 0.0467 | 0.0354 | 0.2245 | 0.2519 | 0.9028 | 0.1734 | 0.2423 | 0.1885 |
| | $\text{BPRH}_{l}$ | 0.0698 | 0.0403 | 0.2762 | 0.3433 | 0.7524 | 0.2137 | 0.3292 | 0.2461 |
| | $\text{BPRH}_{fl}$ | 0.0701 | 0.0402 | 0.2762 | 0.3435 | 0.7547 | 0.2138 | 0.3292 | 0.2462 |
| | $\text{BPRH}''_{vl}$ | 0.0744 | 0.0413 | 0.2806 | 0.3453 | 0.9109 | 0.2241 | 0.3430 | 0.2603 |
| | $\text{BPRH}'_{vl}$ | 0.0769 | 0.0449 | 0.2954 | 0.3488 | 0.9263 | 0.2327 | 0.3493 | 0.2657 |
| | $\text{BPRH}_{vl}$ | 0.0792 | 0.0471 | 0.3018 | 0.3516 | 0.9442 | 0.2462 | 0.3611 | 0.2759 |
| | $\text{BPRH}_{vld}$ | 0.0801 | 0.0474 | 0.3026 | 0.3530 | 0.9465 | 0.2487 | 0.3679 | 0.2783 |

*auxiliary-target correlation* made by *filtered like* is noteworthy in the scenario where social networks are built (i.e., Sobazaar) but almost invisible in the scenario without social network (i.e., Tmall and Xing).

- We also notice that the performance of $\text{BPRH}_v$ is better than that of $\text{BPRH}_l$ and $\text{BPRH}_{fl}$ on Sobazaar; whereas $\text{BPRH}_l$ and $\text{BPRH}_{fl}$ outperform $\text{BPRH}_v$ on Tmall and Xing. According to Figure 3, we know that the overall *auxiliary-target correlation* of *view* is higher than that of *like* and *filtered like* on Sobazaar, whereas an opposite situation is observed on Tmall and Xing. These observations strongly support the conclusion that the *auxiliary action* with higher correlation with the *target action* is more effective to enhance the recommendation performance.

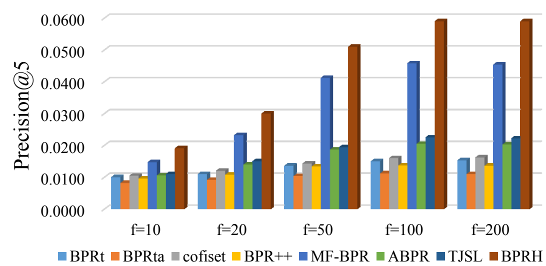
- By generating item-sets based on the *co-selection* scheme proposed in Section 4, $\text{BPRH}_{vl}$ performs better than $\text{BPRH}'_{vl}$ that randomly selects items to build the item-sets, demonstrating the effectiveness of our *co-select* based item-sets generating strategy.

- On all three datasets, $\text{BPRH}_{vl}$ incorporating both *view* and *like* outperforms $\text{BPRH}_v$, $\text{BPRH}_l$ and $\text{BPRH}_{fl}$, which suggests that better recommendation performance can be generated by appropriately integrating both types of *auxiliary action*; Furthermore, $\text{BPRH}_{vld}$ performs slightly better than $\text{BPRH}_{vl}$ on Xing, indicating that *negative action* is also effective to help further improve recommendation accuracy.

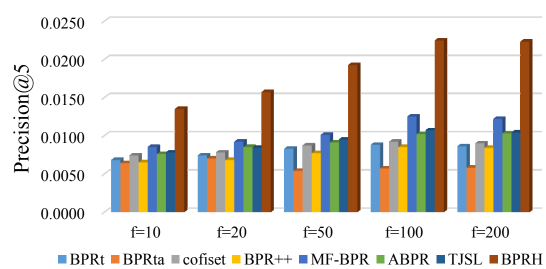
**Comparative Results.** Table 6 presents the performance of all the comparison methods on the three real-world datasets with $f = 50$, where the best performance is highlighted in bold; the best performance of methods proposed by others is marked by 'underline'; the column 'Improve' indicates the relative improvements that our best method achieves w.r.t. the 'underline' results. Besides, we further compare the performance of all comparison methods in terms of precision@5 and AUC with different dimensions of latent factors, i.e.,

Table 6: Recommendation performance of all comparison methods on the three real-world datasets ($f = 50$).

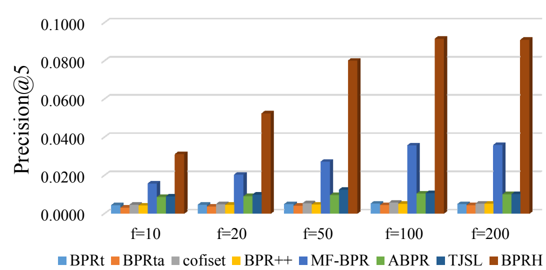
| Dataset | Method | Pre@5 | Pre@10 | Rec@5 | Rec@10 | AUC | MAP | NDCG | MRR |
|---|---|---|---|---|---|---|---|---|---|
| Sobazaar | Random | 0.0014 | 0.0012 | 0.0050 | 0.0012 | 0.6943 | 0.0049 | 0.1142 | 0.0060 |
| | PopRank | 0.0110 | 0.0077 | 0.0399 | 0.0556 | 0.7899 | 0.0274 | 0.1515 | 0.0353 |
| | $\text{BPR}_t$ | 0.0136 | 0.0084 | 0.0369 | 0.0478 | 0.7559 | 0.0251 | 0.1426 | 0.0374 |
| | $\text{BPR}_{ta}$ | 0.0104 | 0.0072 | 0.0322 | 0.0390 | 0.6842 | 0.0221 | 0.1311 | 0.0305 |
| | CoFiSet | 0.0143 | 0.0087 | 0.0372 | 0.0494 | 0.7603 | 0.0277 | 0.1501 | 0.0381 |
| | BPR++(N) | 0.0134 | 0.0078 | 0.0357 | 0.0470 | 0.7475 | 0.0248 | 0.1400 | 0.0314 |
| | MF-BPR | <u>0.0412</u> | <u>0.0237</u> | <u>0.1554</u> | <u>0.2193</u> | <u>0.8753</u> | <u>0.1020</u> | <u>0.2232</u> | <u>0.1231</u> |
| | ABPR | 0.0187 | 0.0103 | 0.0532 | 0.0645 | 0.7826 | 0.0327 | 0.1823 | 0.0504 |
| | TJSL | 0.0194 | 0.0121 | 0.0547 | 0.0651 | 0.7921 | 0.0353 | 0.1965 | 0.0521 |
| | BPRH | **0.0510** | **0.0355** | **0.2005** | **0.2704** | **0.9265** | **0.1378** | **0.2807** | **0.1504** |
| | **Improve** | 24% | 50% | 29% | 23% | 6% | 35% | 26% | 22% |
| Tmall | Random | 0.0003 | 0.0003 | 0.0007 | 0.0015 | 0.5202 | 0.0013 | 0.0957 | 0.0018 |
| | PopRank | 0.0030 | 0.0026 | 0.0089 | 0.0153 | 0.5918 | 0.0082 | 0.1142 | 0.0120 |
| | $\text{BPR}_t$ | 0.0083 | 0.0055 | 0.0213 | 0.0285 | 0.6051 | 0.0169 | 0.1131 | 0.0275 |
| | $\text{BPR}_{ta}$ | 0.0054 | 0.0037 | 0.0162 | 0.0197 | 0.5812 | 0.0118 | 0.1002 | 0.0193 |
| | CoFiSet | 0.0087 | 0.0060 | 0.0217 | 0.0301 | 0.6322 | 0.0172 | 0.1154 | 0.0281 |
| | BPR++(N) | 0.0077 | 0.0053 | 0.0214 | 0.0281 | 0.6215 | 0.0169 | 0.1122 | 0.0276 |
| | MF-BPR | <u>0.0101</u> | 0.0065 | <u>0.0323</u> | <u>0.0468</u> | <u>0.8866</u> | <u>0.0248</u> | <u>0.1486</u> | 0.0309 |
| | ABPR | 0.0091 | 0.0061 | 0.0256 | 0.0353 | 0.7339 | 0.0201 | 0.1234 | 0.0297 |
| | TJSL | 0.0095 | <u>0.0073</u> | 0.0277 | 0.0382 | 0.7565 | 0.0213 | 0.1288 | <u>0.0313</u> |
| | BPRH | **0.0192** | **0.0165** | **0.0668** | **0.1121** | **0.9295** | **0.0486** | **0.1904** | **0.0624** |
| | **Improve** | 91% | 125% | 107% | 140% | 5% | 96% | 28% | 99% |
| Xing | Ramdom | 0.0000 | 0.0000 | 0.0002 | 0.0003 | 0.5432 | 0.0010 | 0.0071 | 0.0007 |
| | PopRank | 0.0021 | 0.0022 | 0.0046 | 0.0068 | 0.6884 | 0.0047 | 0.0888 | 0.0045 |
| | $\text{BPR}_t$ | 0.0050 | 0.0038 | 0.0087 | 0.0122 | 0.6918 | 0.0096 | 0.0957 | 0.0097 |
| | $\text{BPR}_{ta}$ | 0.0041 | 0.0032 | 0.0068 | 0.0083 | 0.6637 | 0.0085 | 0.0886 | 0.0091 |
| | CoFiSet | 0.0054 | 0.0042 | 0.0088 | 0.0143 | 0.7160 | 0.0101 | 0.1027 | 0.0142 |
| | BPR++(N) | 0.0048 | 0.0035 | 0.0083 | 0.0121 | 0.6754 | 0.0097 | 0.0857 | 0.0096 |
| | MF-BPR | <u>0.0272</u> | <u>0.0180</u> | <u>0.1185</u> | <u>0.1523</u> | <u>0.8294</u> | <u>0.0940</u> | <u>0.1704</u> | <u>0.1025</u> |
| | ABPR | 0.0097 | 0.0072 | 0.0242 | 0.0587 | 0.7924 | 0.0313 | 0.1611 | 0.0321 |
| | TJSL | 0.0125 | 0.0089 | 0.0276 | 0.0624 | 0.7981 | 0.0343 | 0.1650 | 0.0331 |
| | BPRH | **0.0801** | **0.0474** | **0.3026** | **0.3530** | **0.9465** | **0.2487** | **0.3679** | **0.2783** |
| | **Improve** | 195% | 164% | 155% | 132% | 14% | 165% | 116% | 171% |




(a) Sobazaar

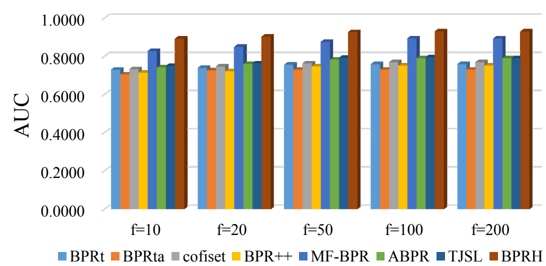



(b) Tmall

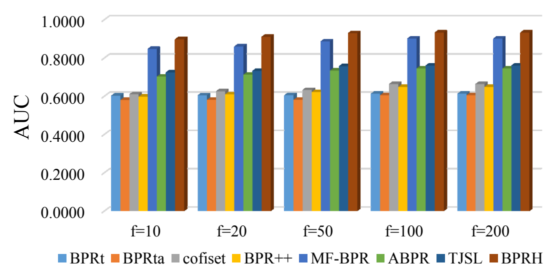



(c) Xing

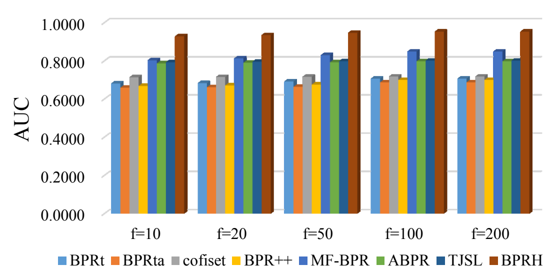
Figure 6: The Pre@5 of all comparison method with different $f$ (dimension of latent factors) on Sobazaar, Tmall and Xing datasets.




(a) Sobazaar




(b) Tmall




(c) Xing

Figure 7: The AUC of all comparison method with different $f$ (dimension of latent factors) on Sobazaar, Tmall and Xing datasets.

$f = \{10, 20, 50, 100, 200\}$. The results are illustrated in Figure 6 and Figure 7, respectively. A number of interesting observations are summarized as follows:

- Unsurprisingly, Random performs the worst among all the other comparison methods. The performance of PopRank is better than Random, while it is generally worse than other BPR based methods, demonstrating the effectiveness of BPR based models. We apply Kendall's $\tau$ coefficient [35] to measure the popularity tendency of comparison methods. The performance of BPRH (0.33) is comparable with the best performance in baselines (0.32, by BPR), which verifies the discussions in Section 3.2.

- Among the BPR based models, the performance of CoFiSet is better than that of $\mathrm{BPR}_t$, indicating that the pairwise assumption over item-sets is more effective than that over a single item. This can be further verified by the fact that $\mathrm{BPRH}''_{vl}$ (over a single item) performs worse than $\mathrm{BPRH}'_{vl}$ and $\mathrm{BPRH}_{vl}$ (over item-sets), as shown in Table 5;

- Both ABPR and TJSL, which take one type of *auxiliary action*, i.e., *view* into consideration, perform better than $\mathrm{BPR}_t$ and CoFiSet that only adopt *target action*. This suggests that *auxiliary action* indeed manages to enhance recommendation performance. On another hand, the fact that $\mathrm{BPR}_t$ performs better than $\mathrm{BPR}_{ta}$ and BPR++(N) indicates that a reasonable fashion is required when *auxiliary action* is integrated into the model. Otherwise, the *auxiliary action* would bring some noise and degenerate the recommendation performance.

- MF-BPR, the integration of multiple types of *auxiliary action*, outperforms ABPR and TJSL which merely consider one type of *auxiliary action*, i.e., *view*, indicating that recommendation can benefit from the combination of different types of *auxiliary action*.

- Our proposed BPRH consistently outperforms MF-BPR across all the metrics on the three real-world datasets. The main reason behind is that

BPRH integrates the *auxiliary action* into the model via a more personalized fashion based on the *auxiliary-target correlation* for each user. The fine-grained assumption also helps model user preference more accurately. Furthermore, BPRH performs the best in different settings of latent factors on all three datasets as shown in Figures 6, 7, demonstrating its effectiveness. As the dimension of latent factors $f$ goes up, the performance of BPRH increases at first and then keeps generally stable when $f >= 100$.

To sum up, BPRH achieves the best performance across all the datasets. The improvements (between BPRH and the best of other methods) across all six metrics are statistically significant (Paired t-test, p-value $< 0.001$), which imply that the recommendation performance can be further boosted by incorporating different types of *auxiliary action* and *negative action* in an appropriate manner.

## 6. Conclusions and Future Work

In this paper, we proposed a Bayesian personalized ranking model for heterogeneous implicit feedback (BPRH) that incorporated *target action*, *auxiliary action*, and *negative action* if available, into a unified model. Specifically, we first assumed that user preference towards items with *target action*, *auxiliary action*, no action and *negative action* were in a descending order. Then we integrated the assumption into a BRP model to build the framework of BPRH. Our method, for the first time, utilized the co-occurrences of heterogeneous actions to quantify the correlation between *auxiliary action* and *target action*, which was further applied to help investigate the difference of user preference among items with different types of actions. Besides, a counterintuitive observation was found that *like* was less correlated with *target action* than *view* in the scenario where a social network was involved, and an adaptive sampling strategy was devised to address the unbalanced correlation among actions. Lastly, in order to better model user preference over item-sets instead of a single item, a *co-selection* based scheme was proposed to build item-sets more efficiently.

Experimental results on three real-world datasets demonstrated the superiority of our proposed model against other counterparts.

A possible limitation of our work might be that we focus on the static correlation between heterogeneous implicit feedback. In fact, user pattern on performing actions as well as user preference should be dynamic. It is also more relevant to practical applications if we investigate how previous actions affect future ones. Thus, for the future work, we plan to investigate the impact of action sequence as well as the temporal information on modeling user preference, to further enhance the recommendation performance.

## 7. Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61401015; the Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education; the Telenor-NTU Joint R&D grant awarded to Dr. Jie Zhang; the National Natural Science Foundation for Young Scientists of China under Grant No. 61702084; the Fundamental Research Funds for the Central Universities under Grant No. N161704001;

## References

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

[2] M. J. Eppler, J. Mengis, The concept of information overload: A review of literature from organization science, accounting, marketing, mis, and related disciplines, Journal of Academic Librarianship 38 (1) (2005) 325–344.

[3] Z. Sun, G. Guo, J. Zhang, Exploiting implicit item relationships for recommender systems, in: Proceedings of the twenty-third Conference on User Modeling, Adaptation and Personalization, Springer, 2015, pp. 252–264.

[4] Y. Koren, Recommender system utilizing collaborative filtering combining explicit and implicit feedback with both neighborhood and latent factor models, US Patents (2011) 8037080 B2.

[5] W. Pan, S. Xia, Z. Liu, X. Peng, Z. Ming, Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks, Information Sciences 332 (C) (2016) 84–93.

[6] J. Lee, D. Lee, Y. C. Lee, W. S. Hwang, S. W. Kim, Improving the accuracy of top- n recommendation using a preference model, Information Sciences 348 (2016) 290–304.

[7] F. Narducci, P. Basile, C. Musto, P. Lops, A. Caputo, M. D. Gemmis, L. Iaquinta, G. Semeraro, Concept-based item representations for a cross-lingual content-based recommendation process, Information Sciences 374 (2016) 15–31.

[8] F. Ortega, A. Hernando, J. Bobadilla, J. H. Kang, Recommending items to group of users using matrix factorization based collaborative filtering, Information Sciences 345 (C) (2016) 313–324.

[9] T. Joachims, L. Granka, B. Pan, H. Hembrooke, G. Gay, Accurately interpreting clickthrough data as implicit feedback, in: Proceedings of the Twenty-eighth International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 154–161.

[10] S. Lee, A. Chandra, D. Jadav, An empirical study on hybrid recommender system with implicit feedback, in: Proceedings of the Twenty-th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Springer International Publishing, 2016, pp. 514–526.

[11] S. Rendle, C. Freudenthaler, Improving pairwise learning for item recommendation from implicit feedback, in: Proceedings of the Seventh ACM International Conference on Web Search and Data Mining, 2014, pp. 273–282.

[12] J. Bauer, A. Nanopoulos, Recommender systems based on quantitative implicit customer feedback, Decision Support Systems 68 (C) (2014) 77–88.

[13] D. Lim, J. McAuley, G. Lanckriet, Top-n recommendation with missing implicit feedback, in: Proceedings of the Ninth ACM Conference on Recommender Systems, 2015, pp. 309–312.

[14] X. Ning, G. Karypis, Slim: Sparse linear methods for top-n recommender systems, in: Proceedings of the Eleventh IEEE International Conference on Data Mining, 2011, pp. 497–506.

[15] D. Yang, T. Chen, W. Zhang, Q. Lu, Y. Yu, Local implicit feedback mining for music recommendation, in: Proceedings of the Sixth ACM Conference on Recommender Systems, 2012, pp. 91–98.

[16] D. Kelly, J. Teevan, Implicit feedback for inferring user preference: a bibliography, The ACM SIGIR Forum 37 (2) (2010) 18–28.

[17] D. W. Oard, J. Kim, Implicit feedback for recommender systems, in: Proceedings of the AAAI Workshop on Recommender Systems, 1998.

[18] B. Loni, R. Pagano, M. Larson, A. Hanjalic, Bayesian personalized ranking with multi-channel user feedback, in: Proceedings of the Tenth ACM Conference on Recommender Systems, 2016, pp. 361–364.

[19] L. Guo, J. Ma, Z. Chen, H. Zhong, Learning to recommend with social contextual information from implicit feedback, Soft Computing 19 (5) (2015) 1351–1362.

[20] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the Eighth IEEE International Conference on Data Mining, 2008, pp. 263–272.

[21] L. Lerche, D. Jannach, Using graded implicit feedback for bayesian personalized ranking, in: Proceedings of the Eighth ACM Conference on Recommender Systems, 2014, pp. 353–356.

[22] W. Pan, L. Chen, Group bayesian personalized ranking with rich interactions for one-class collaborative filtering, Neurocomputing 207 (2016) 501–510.

[23] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: proceedings of the twenty-fifth Conference on Uncertainty in Artificial Intelligence, 2009, pp. 452–461.

[24] G. Guo, J. Zhang, D. Thalmann, A simple but effective method to incorporate trusted neighbors in recommender systems, in: Proceedings of the twenty-second Conference on User Modeling, Adaptation and Personalization, Springer, 2012, pp. 114–125.

[25] W. Pan, M. Liu, Z. Ming, Transfer learning for heterogeneous one-class collaborative filtering, Intelligent Systems IEEE 31 (4) (2016) 1–1.

[26] W. Pan, H. Zhong, C. Xu, Z. Ming, Adaptive bayesian personalized ranking for heterogeneous implicit feedbacks, Knowledge-Based Systems 73 (2015) 173–180.

[27] J. Wang, L. Lin, P. Yu, H. Zhang, Brand recommendation leveraging heterogeneous implicit feedbacks, in: Asia-Pacific World Congress on Computer Science and Engineering, 2015, pp. 1–6.

[28] J. Wang, L. Lin, H. Zhang, J. Tu, Confidence-learning based collaborative filtering with heterogeneous implicit feedbacks, in: Asia-Pacific Web Conference, Springer, 2016, pp. 444–455.

[29] H. Qiu, G. Guo, J. Zhang, Z. Sun, H. T. Nguyen, Y. Liu, Tbpr: Trinity preference based bayesian personalized ranking for multivariate implicit feedback, in: Proceedings of the Twenty-Fourth Conference on User Modeling Adaptation and Personalization, 2016, pp. 305–306.

[30] W. Pan, L. Chen, Cofiset: Collaborative filtering via learning pairwise preferences over item-sets, in: Proceedings of the Siam International Conference on Data Mining, 2013.

[31] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: Proceedings of the Eighth IEEE International Conference on Data Mining, 2008, pp. 502–511.

[32] T. Zhao, J. McAuley, I. King, Leveraging social connections to improve personalized ranking for collaborative filtering, in: Proceedings of the Twenty-third ACM International Conference on Information and Knowledge Management, 2014, pp. 261–270.

[33] Y. Li, J. Hu, C. Zhai, Y. Chen, Improving one-class collaborative filtering by incorporating rich user information, in: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, 2010, pp. 959–968.

[34] S. Kabbur, X. Ning, G. Karypis, Fism: factored item similarity models for top-n recommender systems, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 659–667.

[35] X. Mao, Q. Li, H. Xie, Y. Rao, Popularity tendency analysis of ranking-oriented collaborative filtering from the perspective of loss function, in: The nineteenth International Conference on Database Systems for Advanced Applications, 2014, pp. 451–465.

[36] A. Herschtal, B. Raskutti, Optimising area under the roc curve using gradient descent, in: Proceedings of the Twenty-first International Conference on Machine Learning, 2004, p. 49.

[37] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, pp. 116–123.

[38] T. N. Hai, T. Almenningen, M. Havig, H. Schistad, A. Kofod-Petersen, H. Langseth, H. Ramampiaro, Learning to rank for personalised fashion recommender systems via implicit feedback, in: Proceedings of the Second International Conference on Mining Intelligence and Knowledge Exploration, 2014, pp. 51–61.

[39] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems 20 (4) (2002) 422–446.

[40] Y. Yue, T. Finley, F. Radlinski, T. Joachims, A support vector method for optimizing average precision, in: Proceedings of the Thirtieth International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 271–278.

[41] C. J. C. Burges, R. Ragno, Q. V. Le, Learning to rank with nonsmooth cost functions, in: Proceedings of the Nineteenth International Conference on Neural Information Processing Systems, MIT Press, 2006, pp. 193–200.