Front.Comput.Sci.
DOI

RESEARCH ARTICLE

TransRec++: Translation-based Sequential Recommendation with Heterogeneous Feedback

Zhuo-Xin ZHAN, Ming-Kai HE, Wei-Ke PAN (✉), Zhong MING

1 National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University
2 College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Abstract Sequential recommendation with heterogeneous feedback (SRHF) is an emerging and important problem in recommender systems, which aims to model users' heterogeneous feedback such as examinations and purchases as well as their sequential information. There are some specific challenges in SRHF, including the heterogeneity challenge from users' different types of behaviors, the correlation challenge from the coupling between the items and the behaviors, and the interpretability challenge of the recommendation mechanism. As a response, we propose a novel translation-based recommendation method called TransRec++ with a "one stone, three birds" strategy (i.e., behavior transition vectors) for the above three challenges. Specifically, we introduce some behavior transition vectors to capture the dynamics of users' heterogeneous behaviors, which model the items and the behaviors in a sequence in an intuitive and unified way, and can be easily applied to deep learning-based algorithms. Moreover, we take some latest preceding items into consideration, and learn a weight for each of them automatically. We then conduct extensive empirical studies on four real-world datasets and find that the behavior transition vectors are very effective in modeling heterogeneous sequential feedback, and our TransRec++ outperforms the baselines significantly in most cases. In addition, we analyze the behavior transition numbers in different data and visualize the learned behavior transition vectors, which provide us more insights of our TransRec++.

Keywords Sequential Recommendation, Heterogeneous Feedback, Translation-based Recommendation

Received month dd, yyyy; accepted month dd, yyyy

E-mail: zhanzhuoxin2018@email.szu.edu.cn, 1910273003@email.szu.edu.cn, {panweike,mingz}@szu.edu.cn

1 Introduction

Nowadays, because of the information overload problem on the Internet, recommender systems are widely used from video recommendation[1] to product recommendation[2]. There are a variety of feedback in recommender systems, such as explicit feedback, implicit feedback, sequential feedback, etc. Among them, modeling of heterogeneous sequential feedback, which contains not only different types of feedback such as examinations and purchases but also the sequential information, is an emerging and important problem receiving more and more attention. Heterogeneous sequential feedback is relatively easy to be collected in a deployed system and is also able to provide more information than the homogeneous feedback, which is thus expected to be helpful in improving the recommendation accuracy.

Adaptive Bayesian personalized ranking (ABPR) [16] is probably the first method modeling heterogeneous implicit feedback, which incorporates two kinds of feedback into an objective function, and assigns a weight that can be learned automatically. Transfer via joint similarity learning (TJSL) [12] designs a new prediction rule which involves two kinds of feedback based on FISM [6]. RBPR [18] and RoToR [15] handle such feedback from a new perspective, i.e., they propose a two-stage framework. In the first stage, they use both the two kinds of feedback to obtain a candidate list of a user's likely-to-purchase items, and then refine the list by

---

[1] https://www.netflix.com/
[2] https://www.amazon.com/

exploiting the purchase feedback in the second stage. However, the aforementioned methods do not take the sequential information into consideration, and therefore cannot model the dynamics of the users' preferences well.

Factorizing personalized Markov chains (FPMC) [20] is a pioneer method for modeling homogenous sequential feedback. It is based on matrix factorization and uses the dot product of (user, target item) and (previous item, target item) to model the user's long-term and short-term preferences, respectively. Factorized sequential prediction with item similarity models (Fossil) [5] makes an improvement on FPMC. Specifically, it combines an item-similarity model (i.e., FISM [6]) with high-order Markov chains, which achieves the state-of-the-art performance. However, these methods only make use of the homogeneous feedback and do not consider the effects of different types of behaviors.

Recommendation with heterogeneous sequential feedback or sequential recommendation with heterogeneous feedback (SRHF) is a new recommendation problem and has rarely been studied. Recommendation on micro behaviors (RIB) [27] exploits an RNN layer and an attention layer to handle the sequential information and heterogeneous feedback, respectively. Behavior-intensive neural network (BINN) [8] feeds the item representations, sequential information and heterogeneous feedback into two LSTM-based modules. The existing methods are mainly based on deep learning [2, 8, 10, 27] and lack good interpretability. Moreover, they model the users' items and behaviors separately, rather than treat them in a holistic manner.

In this paper, we design some novel behavior transition vectors and propose a novel solution named TransRec++ for SRHF. Specifically, our behavior transition vectors can model the dynamics of the users' behaviors, one merit of which is that it captures the users' heterogeneous feedback and sequential information in an intuitive and unified way. We then notice that a user's behavior transition can be affected by some latest preceding items, so we take a few preceding items into consideration and assign a weight to each of them which is learned automatically. We conduct extensive experiments on four real-world datasets, and the results showcase the effectiveness of our TransRec++.

We summarize our main contributions as follows: i) we study an emerging and important problem (i.e., sequential recommendation with heterogeneous feedback, SRHF) from a new perspective (i.e., preference transition between heterogeneous feedback); ii) we propose a novel translation-based sequential recommendation method called TransRec++ that captures four different types of transitions, including from examination/purchase to examination/purchase, which is of good interpretability; iii) we conduct extensive empirical studies to show the effectiveness of our TransRec++ over the state-of-the-art translation-based and non-translation-based methods; iv) we study the behavior transition numbers in different data and visualize the learned behavior transition vectors, and obtain more insights of our solution; and v) we generalize our TransRec++ to problems with more than two types of behaviors, and apply the behavior transition vectors to some deep learning-based methods, which demonstrates the flexibility and generality of our solution.

The rest of the paper is organized as follows. In Section 2, we discuss some related work about recommendation with homogenous and heterogeneous feedback. In Section 3, we define our studied problem, discuss the challenges and present an overview of our solution. In Section 4, we describe our solution in detail, and discuss the time complexity and the generalization of our TransRec++ for cases with more than two types of behaviors, as well as its application to deep learning-based methods. In Section 5, we conduct extensive experiments and analyze the results to show the effectiveness and generality of our TransRec++. In Section 6, we conclude our work with some interesting future directions.

2 Related Work

2.1 Recommendation with Homogeneous Feedback

2.1.1 Homogeneous explicit feedback

Recommendation algorithms handling homogeneous explicit feedback exploit users' numerical ratings on their interacted items in order to predict their scores on the un-interacted items. Generally, these algorithms can be classified into two categories, i.e., neighborhood-based methods and matrix factorization-based methods. Neighborhood-based methods [21] first calculate the similarities between users (or items) to obtain each user's (or each item's) neighbors, and then predict the ratings based on the information of the similar users (or items). Matrix factorization-based methods [13, 17] represent a user's numerical rating to an item as a dot product of the user-specific vector and the item-specific vector, and then learn the latent vectors by minimizing the error between the predicted score and the true score.

2.1.2 Homogeneous implicit feedback

In real-world applications, implicit feedback such as purchases are usually collected more abundantly than explicit feed-

Table 1 Summary of some shallow learning-based methods in modeling different types of feedback.

	Homogeneous Feedback	Heterogeneous Feedback
Explicit	PMF [17], etc.	TMF [14], PAT [9], etc.
Implicit	BPR [19], FISM [6], etc.	RBPR [18], RoToR [15], etc.
Sequential	FPMC [20], Fossil [5], TransRec [4], etc.	TransRec++ (**proposed in this paper**)

back. Bayesian personalized ranking (BPR) [19] makes use of a pairwise preference assumption where a user prefers an interacted item to an un-interacted one. Factored item similarity model (FISM) [6] is based on a pointwise preference assumption where a user likes an interacted item and dislikes an un-interacted one or a pairwise preference assumption similar to that of BPR, and learns the item-item similarities in the prediction rule.

2.1.3 Homogeneous sequential feedback

Sequential recommendation algorithms utilize the sequential information in order to predict a user's next likely-to-interact item. Factorizing personalized Markov chains (FPMC) [20] combines Markov chains for users' short-term sequential preferences with a general matrix factorization method for users' long-term preferences. Factorized sequential prediction with item similarity models (Fossil) [5] fuses a similarity model (i.e., FISM) with high-order Markov chains so as to address the sparsity issue and capture the sequential information in a unified framework. Translation-based recommendation (TransRec) [4] draws inspirations from knowledge graph [22]. It embeds items into a transition space and models users as translation vectors in order to capture the users' dynamics.

2.2 Recommendation with Heterogeneous Feedback

2.2.1 Heterogeneous explicit feedback

Heterogeneous explicit feedback refers to a combination of two different types of explicit feedback such as numerical ratings and binary ratings. SVD++ [7] is a seminal work for combining heterogeneous feedback such as explicit feedback and implicit feedback in one single prediction rule. Following SVD++, transfer by mixed factorization (TMF) [14] and preference-aware transfer (PAT) [9] address such heterogeneous feedback from a transfer learning perspective, which takes the numerical ratings as the target data and the binary ratings as the auxiliary data, rather than simply leverage the two kinds of ratings in an integrative manner only, so as to well capture the feedback-dependent effect.

2.2.2 Heterogeneous implicit feedback

In heterogeneous implicit feedback, users' two kinds of implicit feedback such as examinations and purchases are given. Adaptive Bayesian personalized ranking (ABPR) [16] designs a BPR-based objective function, and learns a confidence on each examination adaptively, aiming to address the uncertainty of the implicit feedback. Transfer via joint similarity learning (TJSL) [12] designs a new prediction rule which involves two kinds of feedback, and learns a similarity between a target item and a purchased item as well as a similarity between a target item and an identified likely-to-prefer examined item. Role-based methods [15, 18] model the heterogeneous feedback of a user via two roles, i.e., an examiner and a purchaser, and predict the users' preferences via a two-stage preference learning approach.

2.2.3 Heterogeneous sequential feedback

There are very few works that model heterogeneous sequential feedback, and almost all of them are deep learning-based methods. Recommendation on micro behaviors (RIB) [27] includes an embedding layer to address the sparsity and high-dimensionality issues, an RNN layer to handle the sequential information, and an attention layer to capture the effects of heterogeneous feedback. Behavior-intensive neural network (BINN) [8] uses a modified item2vec [1] called w-item2vec to obtain the item representations, and then feed the item embedding, sequential information and heterogeneous feedback into two LSTM-based modules. We can see that the recommendation mechanisms in these methods are difficult to be understood. Moreover, they model the items and the behaviors in a same sequence separately rather than treat them in a unified way.

From the above discussions, we can see that most methods are based on shallow learning (i.e., factorization-based or translation-based methods), which is associated with the merit of good interpretability. Interestingly, methods for modeling heterogeneous sequential feedback are all based on deep learning, which motivates us to focus on shallow learning and design a translation-based method for SRHF. We briefly summarize the shallow learning-based methods for modeling different types of feedback in Table 1.

3 Sequential Recommendation with Heterogeneous Feedback

3.1 Problem Definition

In sequential recommendation with heterogeneous feedback (SRHF), we have a set of interaction sequences $\mathcal{S}_u = \{(i_u^1, b(i_u^1)), (i_u^2, b(i_u^2)), \ldots, (i_u^{|\mathcal{S}_u|}, b(i_u^{|\mathcal{S}_u|}))\}$ for every user $u \in \mathcal{U}$, where i_u^t represents the tth item in $\mathcal{S}_u$ and $b(i_u^t) \in \{examination, purchase\}$ is the behavior of user u on item i_u^t. We illustrate the studied problem in Fig. 1. As shown in Fig. 1, our goal is to predict the user's next likely-to-purchase item, i.e., $(i_u^{|\mathcal{S}_u|+1}, b(i_u^{|\mathcal{S}_u|+1}) = purchase)$, for each user u. We list the notations and their corresponding explanations used in the paper in Table 2.

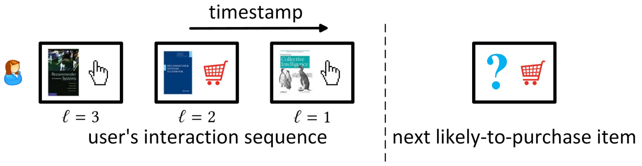

Fig. 1 Illustration of sequential recommendation with heterogeneous feedback (SRHF), where a user's interaction sequence of heterogeneous feedback (i.e., examinations and purchases) are exploited in order to recommend a next likely-to-purchase item.

3.2 Challenges

In order to solve the SRHF problem effectively, we have to address the following challenges. (i) The *heterogeneity* challenge. Users' preferences beneath different types of behaviors such as examinations and purchases are often different, for which a traditional recommendation method may fail to capture such differences. (ii) The *correlation* challenge. Items and behaviors are often highly correlated in a user's interaction sequence, for which treating the sequence of items and the sequence of behaviors of a same user separately in existing works may not be sufficient. (iii) The *interpretability* challenge. Users usually expect to understand the recommendation mechanism from the perspective of human being, for which existing deep learning-based methods are often too complex.

3.3 Overall of Our Solution

We address the aforementioned three challenges by proposing a novel solution with a series of behavior transition vectors, including from examination to examination (E2E), from examination to purchase (E2P), from purchase to examination (P2E), and from purchase to purchase (P2P), which is actually a "one stone, three birds" strategy. Specifically, the behavior transition vectors treat the items and their behaviors as a whole ("correlation") and model the changes of the users' different types of behaviors ("heterogeneity") in an intuitive way via E2E, E2P, P2E and P2P ("interpretability").

Technically, we integrate the high-order behavior transition vectors into a translation-based sequential recommendation model, i.e., TransRec [4], and propose a novel solution called TransRec++ for the SRHF problem. Specifically, our model first embeds items into a so-called transition space, and then considers the behavior transitions between a target item and its L latest preceding items, which is illustrated in Fig. 2. Compared with the existing methods, our model takes users' sequences of items and behaviors into consideration in a unified way and captures the transition relationships in a natural and interpretable way. Moreover, our model is a shallow learning-based method, which means that it consumes less computing resource than the counterparts, i.e., the deep learning-based methods.

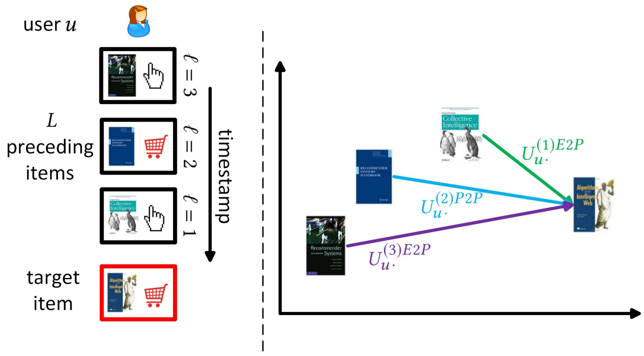

Fig. 2 Illustration of our TransRec++, which first embeds items into vectors in a transition space and then learns user-specific behavior transition vectors with different orders $1 \le \ell \le L$.

4 Our Solution

4.1 Translations

We tackle the heterogeneous feedback transition relationships by introducing high-order behavior transition vectors for each user u, i.e., $U_{u\cdot}^{(\ell)E2E}$, $U_{u\cdot}^{(\ell)E2P}$, $U_{u\cdot}^{(\ell)P2E}$ and $U_{u\cdot}^{(\ell)P2P}$. They represent a user u's ℓ-order examination-to-examination, examination-to-purchase, purchase-to-examination and purchase-to-purchase transition vectors, respectively. For instance, in Fig. 2, we can see that user u examines the

Table 2 Some notations and their explanations used int the paper.

n	the number of users
m	the number of items
$\mathcal{U}, \lvert\mathcal{U}\rvert = n$	the whole set of users
$\mathcal{I}, \lvert\mathcal{I}\rvert = m$	the whole set of items
$u \in \{1, 2, \ldots, n\}$	user ID
$j \in \{1, 2, \ldots, m\}$	item ID
$b(i_u^t)$	the behavior on item i_u^t, $b(i_u^t) \in \{examination, purchase\}$
$\mathcal{S}_u = \{(i_u^t, b(i_u^t))\}$	the sequence of (item, behavior) pairs with heterogeneous feedback of user u
i_u^t	the tth item in $\mathcal{S}_u$
$\mathcal{R}$	the whole set of observed (user, item, behavior) tuples
$V_{i_u^t\cdot} \in \mathbb{R}^{1\times d}$	the embedding vector of item i_u^t
$p_i \in \mathbb{R}$	the bias of item i
Θ	the set of model parameters
$\lvert\lvert\mathbf{x}\rvert\rvert_2^2 = \mathbf{x}^T\mathbf{x}$	the squared L_2 norm (i.e., the square of the length of $\mathbf{x}$)
γ	the learning rate
$\alpha_v, \alpha_u, \beta_b, \beta_\eta$	the regularization parameter
T	the iteration number in the algorithm
$U_{u\cdot} \in \mathbb{R}^{1\times d}$	the global translation vector of user u
$U_{u\cdot}^{(\ell)E2E} \in \mathbb{R}^{1\times d}$	the ℓ-order examination-to-examination transition vector of user u
$U_{u\cdot}^{(\ell)E2P} \in \mathbb{R}^{1\times d}$	the ℓ-order examination-to-purchase transition vector of user u
$U_{u\cdot}^{(\ell)P2E} \in \mathbb{R}^{1\times d}$	the ℓ-order purchase-to-examination transition vector of user u
$U_{u\cdot}^{(\ell)P2P} \in \mathbb{R}^{1\times d}$	the ℓ-order purchase-to-purchase transition vector of user u
$\tilde{U}_{u\cdot}^{(\ell)i_u^t}$	the overall translation vector of user u within ℓ step(s) to the target item i_u^t
L	the number of latest preceding items concerned
$\eta_\ell \in \mathbb{R}$	the ℓ-order weight shared by all users
$\eta_\ell^u \in \mathbb{R}$	the ℓ-order weight of user u

black book 3 steps before she purchases the last book, so the transition vector is $U_{u\cdot}^{(3)E2P}$ as shown in purple in the right part of the figure.

Furthermore, we introduce a vector $U_{u\cdot}$ in order to capture a user's global preference. Finally, the overall translation vector of user u within ℓ step(s) to the target item i_u^t, defined as $\tilde{U}_{u\cdot}^{(\ell)i_u^t}$, is formulated as follows,

$$\tilde{U}_{u\cdot}^{(\ell)i_u^t} = U_{u\cdot} + U_{u\cdot}^{(\ell)b(i_u^{t-\ell})2b(i_u^t)}, \tag{1}$$

where $b(i_u^{t-\ell})$ and $b(i_u^t)$ denote the types of feedback (i.e., behaviors) on item $i_u^{t-\ell}$ and item i_u^t, respectively.

We aim to make the distance between our predicted item and the true target item in the transition space as close as possible. In this way, an item in a user's sequence can naturally achieve a transition to a future item in ℓ steps by adding the user's overall translation vector, which is formulated as follows,

$$V_{i_u^{t-\ell}\cdot} + \tilde{U}_{u\cdot}^{(\ell)i_u^t} \approx V_{i_u^t\cdot}, \ell = 1, 2, \ldots, L, \tag{2}$$

where $V_{i_u^{t-\ell}\cdot}$ and $V_{i_u^t\cdot}$ are the embedding vectors of item $i_u^{t-\ell}$ and item i_u^t, respectively.

4.2 Prediction Rule and Objective Function

According to Eq.(2), the preference $\hat{r}_{ui_u^t}$ for user u on item i_u^t is defined as follows,

$$\hat{r}_{ui_u^t} \propto -\sum_{\ell=1}^{L} \lvert\lvert V_{i_u^{t-\ell}\cdot} + \tilde{U}_{u\cdot}^{(\ell)i_u^t} - V_{i_u^t\cdot}\rvert\rvert_2^2. \tag{3}$$

Moreover, we find that different users have different shopping habits, so we assign a user-specific weight η_ℓ^u together with a global weight η_ℓ to each distance term. We follow TransRec [4] and introduce an item bias $p_{i_u^t}$ to suppress popular items. Finally, the predicted preference $\hat{r}_{ui_u^t}$ for user u on item i_u^t is as follows,

$$\hat{r}_{ui_u^t} = p_{i_u^t} - \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u) \lvert\lvert V_{i_u^{t-\ell}\cdot} + \tilde{U}_{u\cdot}^{(\ell)i_u^t} - V_{i_u^t\cdot}\rvert\rvert_2^2. \tag{4}$$

From Eq.(4), we can see that user u's preference on item i_u^t is related to the item's L latest preceding items and the user-specific weight assigned to the corresponding distance.

Compared with TransRec [4], we extend its user translation vector to a user's overall translation vector, which is composed of a global preference translation vector and a behavior transition vector to capture the dynamics between different types of behaviors. Moreover, we take more preceding items of a target item into consideration and assign some user-specific weights to different terms of distances.

We follow previous works and adopt the S-BPR loss function [20]. Specifically, for a tuple (u, i_u^t, j), $j \in \mathcal{I} \backslash \mathcal{S}_u$, we have the following objective function to be maximized,

$$\max \sum_{(u, i_u^t, j), j \in \mathcal{I} \backslash \mathcal{S}_u} \ln \sigma(\hat{r}_{ui_u^t} - \hat{r}_{uj}) - \text{Reg}(\Theta), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, Θ is the set of model parameters to be updated, and $\text{Reg}(\Theta)$ is the L_2 regularization term w.r.t. Θ.

We adopt the stochastic gradient ascent (SGA) algorithm for updating the model parameters. For each parameter $\theta \in \Theta$, we have the update rule

$$\theta = \theta + \gamma \nabla \theta, \tag{6}$$

where $\gamma > 0$ is the learning rate.

4.3 Gradients

For $\theta \in \Theta$, the gradient is computed as follows,

$$\nabla \theta = \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) \nabla (\hat{r}_{ui_u^t} - \hat{r}_{uj}) - \beta_\theta \theta. \tag{7}$$

Therefore, we have the gradients,

$$\nabla p_{i_u^t} = \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) - \beta_p p_{i_u^t}, \tag{8}$$

$$\nabla p_j = -\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) - \beta_p p_j, \tag{9}$$

$$\begin{aligned} \nabla V_{i_u^t \cdot} = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[2 \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)(V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot})] \\ & -\alpha_v V_{i_u^t \cdot}, \end{aligned} \tag{10}$$

$$\begin{aligned} \nabla V_{i_u^{t-\ell} \cdot} = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[2(\eta_\ell + \eta_\ell^u)((V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{j \cdot}) \\ & -(V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot}))] \\ & -\alpha_v V_{i_u^{t-\ell} \cdot}, \ell = 1, ..., L, \end{aligned} \tag{11}$$

$$\begin{aligned} \nabla V_{j \cdot} = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[-2 \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)(V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{j \cdot})] \\ & -\alpha_v V_j, \end{aligned} \tag{12}$$

$$\begin{aligned} \nabla U_{u \cdot} = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[2 \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{j \cdot}) \\ & -(V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot}))] - \alpha_u U_{u \cdot}, \end{aligned} \tag{13}$$

$$\begin{aligned} \nabla U_{u \cdot}^{(\ell) b(i_u^{t-\ell}) 2 b(i_u^t)} = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[2(\eta_\ell + \eta_\ell^u)((V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{j \cdot}) \\ & -(V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot}))] \\ & -\alpha_u U_{u \cdot}^{(\ell) b(i_u^{t-\ell}) 2 b(i_u^t)}, \ell = 1, ..., L, \end{aligned} \tag{14}$$

$$\begin{aligned} \nabla \eta_\ell = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})(||V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_j||_2^2 \\ & -||V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot}||_2^2) \\ & -\beta_\eta \eta_\ell, \ell = 1, ..., L, \end{aligned} \tag{15}$$

$$\begin{aligned} \nabla \eta_\ell^u = {} & \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})(||V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_j||_2^2 \\ & -||V_{i_u^{t-\ell} \cdot} + \tilde{U}_{u \cdot}^{(\ell) i_u^t} - V_{i_u^t \cdot}||_2^2) \\ & -\beta_\eta \eta_\ell^u, \ell = 1, ..., L. \end{aligned} \tag{16}$$

From the perspective of update formulas, our model updates L latest preceding items' embedding vectors, i.e., $\{V_{i_u^{t-\ell} \cdot} | \ell = 1, \ldots, L\}$, while TransRec only updates one, i.e., $V_{i_u^{t-1} \cdot}$. Moreover, our model updates different behavior transition vectors according to the user's behavior sequence and the user-specific weights on different steps, which is again different from that of TransRec.

4.4 The Learning Algorithm

We depict the SGA-based algorithm of our TransRec++ in Alg. 1. Firstly, we initialize the model parameters Θ. The outer loop ranging from 1 to T contains two parts: the inner loop part samples a (user, item) pair, L preceding items and a negative item to update the model parameters, before re-normalizing the involved item embedding vectors to restrain them in a unit L_2 ball. Notice that when the HR@k performance does not improve within 300 iterations, the training will be stopped.

Algorithm 1 The algorithm of TransRec++.

```
Initialize the model parameters $\Theta$
for $iter = 1, ..., T$ do
    for $iter2 = 1, ..., |\mathcal{R}|$ do
        Randomly pick up a pair $(u, i_u^t) \in \mathcal{R} \backslash \{i_u^1, ..., i_u^L\}$ (ensure $i_u^t$ has $L$ preceding items)
        Take $L$ latest preceding items $i_u^{t-L}, ..., i_u^{t-1}$
        Randomly pick up a negative item $j \in \mathcal{I} \backslash \mathcal{S}_u$
        Calculate the gradients via Eqs.(8-16)
        Update the model parameters via Eq.(6)
        Re-normalize the item embedding vector $V_{*,\cdot} = \frac{V_{*,\cdot}}{\max(1, \|V_{*,\cdot}\|)}$ for $V_{i_u^{t-L}\cdot}$, ... $V_{i_u^{t-1}\cdot}$, $V_{i_u^t\cdot}$ and $V_{j\cdot}$
```

4.5 Time Complexity Analysis

In the training phase, the training time of our TransRec++ is roughly $5L$ times as that of TransRec because of the gradient calculation and parameter update of four kinds of behavior transition vectors, i.e., (E2E, E2P, P2E and P2P) × L steps + L preceding item embedding vectors. In the test phase, the test time is about L times as that of TransRec because of the calculation of the distances w.r.t. the newly added $L - 1$ preceding items' embedding vectors. In empirical studies, however, we find that a larger L usually leads to faster convergence as it can capture the sequential information more sufficiently, which means that the running time of our TransRec++ is actually shorter than the above analysis.

4.6 Discussions

When only purchases in the heterogeneous feedback are preserved and $L = 1$, all the user-related vectors in our TransRec++ play the same role as the user translation vector in TransRec [4]. Hence, our TransRec++ embodies TransRec as a special case.

More importantly, our TransRec++ is quite flexible as it can tackle complicated sequential recommendation problems with more than two types of behaviors because we can simply include more behavior transition vectors. For example, when the users' "add-to-cart" behaviors are involved, we can generalize our TransRec++ by including some additional behavior transition vectors, i.e., $U_{u\cdot}^{(\ell)E2A}$, $U_{u\cdot}^{(\ell)P2A}$, $U_{u\cdot}^{(\ell)A2E}$, $U_{u\cdot}^{(\ell)A2P}$, $U_{u\cdot}^{(\ell)A2A}$, where "A" denotes the "add-to-cart" behavior. To demonstrate this, we conduct experiments to generalize our TransRec++ to solve sequential recommendation with four types of behaviors in Section 5.5.6.

Our behavior transition vectors are also of good generality as they can be easily extended to deep learning-based methods to achieve better recommendation accuracy. We also confirm this by including empirical studies in Section 5.5.7.

5 Experiments

5.1 Datasets

We conduct experiments on four public datasets, including MovieLens 1M (ML1M), Rec15, User Behavior (UB) and Tmall. The four datasets are all constituted by a series of (user, item, behavior, timestamp) quadruples, which can be considered as users' behavior-aware interaction sequences with items.

ML1M. MovieLens 1M (ML1M) [3] is a pubic dataset collected by the researchers from GroupLens. It contains about 1 million five-point scale ratings given by 6,040 users to 3,952 items with timestamp information from April 2000 to February 2003. In order to simulate heterogenous feedback in S-RHF, we follow a previous work [15], and take the users' 5-score ratings as purchases and the others as examinations.

Rec15. Rec15 is a real dataset provided by RecSys Challenge 2015[3], which includes six months of activities of a big e-commerce business in Europe. It contains 36,917 users' examinations and purchases on 52,739 items with the timestamp information.

UB. User Behavior (UB)[4] is a real dataset collected by Taobao.com. It contains 100,150,807 behavior-aware interactions between 987,994 users and 4,162,024 items during November 25 and December 03, 2017. We keep the examinations and purchases in the experiments.

Tmall. Tmall[5] contains 424,170 users' shopping logs in the past six months before and on the "Double 11" day on Tmall.com in 2015. In order to avoid the users' casual shopping behaviors affected by the promotion activities, we remove all the data on the "Double 11" day. Similar to UB, we keep users' examinations and purchases.

We preprocess the four datasets as follows: i) we discard the cold-start items purchased fewer than 5 times in ML1M and Rec15, 10 in UB, and 20 in Tmall; ii) we discard the cold-start users whose number of purchases is smaller than 10 in Tmall and 5 in the others; iii) we keep the first (user, item, behavior) triple and discard the others if it occurs more than once; and iv) we delete some examinations at the end of the interaction sequence of each user u to ensure that the last

---

[3] `https://recsys.acm.org/recsys15/challenge/`

[4] `https://tianchi.aliyun.com/dataset/dataDetail?dataId=649`

[5] `https://tianchi.aliyun.com/dataset/dataDetail?dataId=42`

two interactions are purchases.

For each dataset, we divide the interaction sequence $\mathcal{S}_u$ of each user u into three parts, i.e., the last purchase interaction $(i_u^{|\mathcal{S}_u|}, purchase)$ as the test set, the penultimate purchase interaction $(i_u^{|\mathcal{S}_u|-1}, purchase)$ as the validation set, and all the rest as the training set.

We put the statistics of the datasets used in the experiments in Table 3.

Table 3 Statistics of the processed datasets used in the experiments.

Dataset	#Users	#Items	#Examinations	#Purchases
ML1M	5,645	2,357	628,892	223,305
Rec15	36,917	9,621	446,442	233,263
UB	20,858	30,793	470,731	136,250
Tmall	17,209	16,176	831,117	240,901

5.2 Evaluation Metrics

We adopt two commonly used ranking-oriented metrics, i.e., Hit Rate@k (HR@k) and NDCG@k, to evaluate the performance of the recommendation methods. HR@k represents the ratio of correct predictions in a top-k recommendation list, and NDCG@k is a position-aware metric that takes the recommended items' positions into consideration.

5.3 Baselines

RBPR [18]. RBPR is a role-based recommendation algorithm for the studied problem without using the sequential information. It models users' different feedback from a new perspective, where each user has two different roles, i.e., a browser and a purchaser, and then derives a two-stage role-based algorithm based on Bayesian personalized ranking (BPR) [19].

RoToR [15]. RoToR is a transfer learning-based recommendation algorithm for modeling examinations and purchases. We adopt one of its four variants, i.e., RoToR(pai., seq.), since this variant is found to perform the best in the real-world dataset [15]. Notice that "seq." means that the method is composed of two consecutive phases where the first phase employs a neighborhood-based algorithm (i.e., ICF [21]) to generate a candidate list of items for each user and the second phase employs a pairwise matrix factorization-based method (i.e., BPR [19]) to re-rank the candidate list. In other words, "seq." does not mean "sequential" in sequential recommendation.

FPMC [20]. FPMC is a classic sequential recommendation algorithm that combines matrix factorization and Markov chains. It applies the inner products $U_{u\cdot}V_{i\cdot}^T$ and $P_{i\cdot}Q_{i\cdot}^T$ to capture the (user, item) interaction and (item, item) interaction, respectively.

Fossil [5]. Fossil is a well-known sequential recommendation method. It combines a factored similarity method with high-order Markov chains, which is found to be better than many factorization-based sequential recommendation algorithms.

TransRec [4]. TransRec is a translation-based method for sequential recommendation, which is the most closely related work to ours. It embeds items into a "transition space" and models users as translation vectors. We follow [4] and use the L_2 distance as the metric.

We can see that RBPR and RoToR model heterogeneous feedback but not the sequential information, and FPMC, Fossil and TransRec model homogeneous sequential feedback. As far as we know, there is no shallow learning-based method that considers both the heterogeneous feedback and the sequential information in SRHF.

5.4 Parameter Configurations

We set $k = 10$ to recommend a top-10 list of items for each user and use HR@10 and NDCG@10 to evaluate the performance of all the methods. We fix the dimension $d = 20$ and apply the same stochastic gradient ascent (SGA) or stochastic gradient descent (SGD) algorithmic framework written in Java for all the methods for fair comparison. Following the original papers [15, 18], we set the size of the candidate list of items as $3K = 30$ for RBPR and RoToR, and the size of neighborhood as 20 for RoToR.

For all the methods, we search the optimal parameters via the HR@10 performance on the validation set. To be specific, we choose the regularization parameter $\alpha_v = \alpha_u = \beta_b = \beta_\eta$ from $\{0.001, 0.01, 0.1\}$, the learning rate γ from $\{0.01, 0.005, 0.002\}$, and the iteration number T from $\{1, 2, ..., 10, 20, \ldots, 1990, 2000\}$, i.e., the step size is 1 and 10 before and after $T = 10$, respectively. If the performance on the validation data does not improve within 300 iterations, the algorithms will quit in order to avoid over-fitting. For Fossil and our TransRec++, we search the order of Markov chain L from $\{1, 2, 3\}$ in a similar way. Notice that the data, code and scripts used in the experiments will be publicly available for reproducibility and further extension once the paper is accepted.

Table 4 Recommendation performance of our TransRec++ and the five baselines on the four datasets.[a,b]

	ML1M		Rec15		UB		Tmall	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
RBPR	0.0778	0.0384	0.3182	0.1594	0.0303	0.0159	0.0156	0.0075
RoToR	0.0764	0.0378	0.3708	0.1940	0.0080	0.0043	0.0313	0.0161
FPMC	0.1086	***0.0510***	***0.3829***	***0.2102***	0.0467	0.0249	0.0352	0.0191
Fossil	**0.1114**	**0.0528**	0.3736	0.2029	0.0508	0.0270	***0.0437***	0.0241
TransRec	0.0852	0.0409	0.3697	0.1928	***0.0589***	***0.0342***	0.0374	***0.0325***
TransRec++	***0.1088***	0.0508	**0.4064**	**0.2209**	**0.0661**	**0.0413**	**0.0593**	**0.0377**

a) Notice that for the sequential recommendation methods (i.e., FPMC, Fossil, TransRec and our TransRec++), we exploit both the examinations and purchases in the training phase and then recommend some un-purchased items for each user in the test phase.

b) The best and second best results are marked in bold and both bold and italic, respectively.

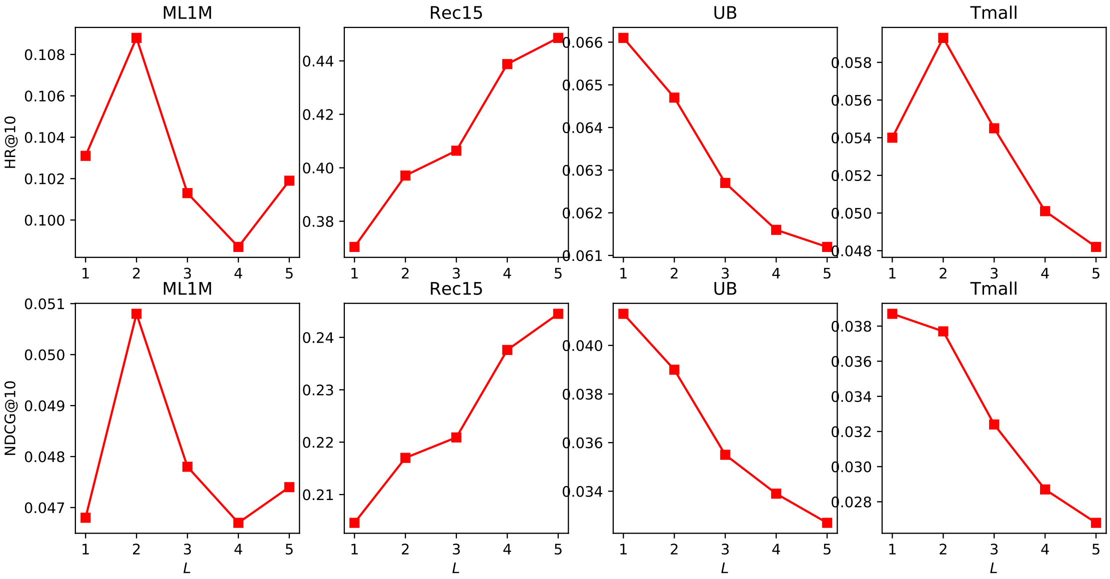

Fig. 3 Recommendation performance of our TransRec++ with different values of L on the four datasets.

5.5 Experimental Results

5.5.1 Main Results

We report the results of our TransRec++ and the baselines in Table 4, from which we have the following observations: (i) Our TransRec++ achieves the best performance among all the methods on all the datasets except on ML1M, which clearly showcases its effectiveness. Nevertheless, on ML1M, our TransRec++ is still very competitive in comparison with the other methods. (ii) In comparison with TransRec, our TransRec++ outperforms it significantly on all the datasets, which demonstrates that the designed high-order behavior transition vectors and user-specific weights well address the heterogeneity and correlation challenges in the studied SRHF problem. (iii) The two HOCCF methods, i.e., RBPR and RoToR, are worse than the sequential recommendation methods, which shows the importance of the sequential information.

5.5.2 Impact of the Number of Concerned Preceding Items

In this section, we adjust $L \in \{1, 2, 3, 4, 5\}$ and study the impact of different values of L, i.e., the number of concerned preceding items, to figure out how it affects the recommendation performance. The results are reported in Fig. 3, from which we can see that our TransRec++ achieves the best overall results when $L = 2$ for the datasets ML1M and Tmall and when $L = 5$ for the dataset Rec15, which demonstrates the effectiveness of the proposed high-order behavior

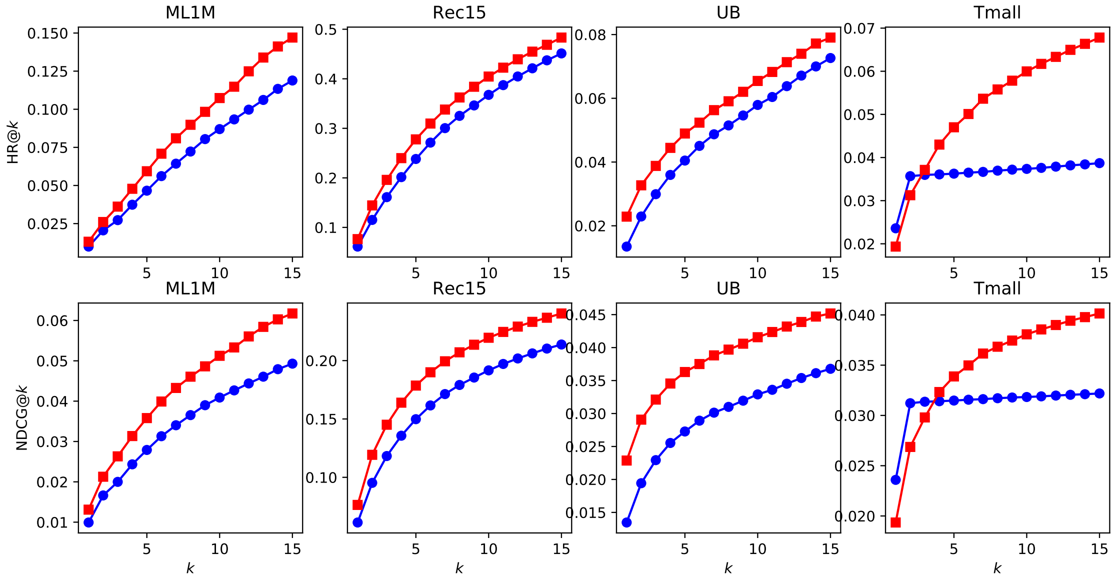

Fig. 4 Top-*k* recommendation performance on the four datasets, where the blue line and the red line represent the results of TransRec and our TransRec++, respectively.

transition vectors. For the dataset UB, we find that our TransRec++ achieves the best performance when $L = 1$, which means that we should choose an appropriate value of L for different datasets.

5.5.3 Results of Top-*k* Recommendation List

In this section, we study the top-k recommendation performance with different values of $k \in \{1, 2, ..., 15\}$, which are shown in Fig. 4. From Fig. 4, we can see that our TransRec++ performs better than the closely related baseline method TransRec in almost all cases, which again clearly showcases the effectiveness of the introduced high-order behavior transition vectors in our TransRec++.

5.5.4 Analyses of the High-order Behavior Transition Relationships

We count the numbers of the ℓ-order behavior transitions, denoted as $\#(\ell)E2E$, $\#(\ell)E2P$, $\#(\ell)P2E$, $\#(\ell)P2P$, $\ell \in \{1, 2, 3\}$, in the training data of each dataset. We report the statistics in Table 5.

Table 5 Statistics of the transition numbers.

	ML1M	Rec15	UB	Tmall
#(1)E2E	500,685	409,165	397,319	676,749
#(2)E2E	494,561	372,035	377,609	662,361
#(3)E2E	489,298	334,999	362,960	651,446
#(1)E2P	124,400	37,168	60,408	141,807
#(2)E2P	126,618	74,129	65,629	143,455
#(3)E2P	127,936	110,976	65,410	141,742
#(1)P2E	123,832	360	56,444	145,181
#(2)P2E	125,567	608	59,161	148,033
#(3)P2E	126,456	760	57,013	146,225
#(1)P2P	86,345	122,261	30,236	56,654
#(2)P2P	82,871	85,265	21,150	49,333
#(3)P2P	80,282	48,385	17,308	46,560

From Table 5, the examination-to-examination transition relationship dominate all the datasets, which results from the larger number of examination behaviors in the datasets. And we can see that as the value of ℓ increases, $\#(\ell)E2E$ drops, because users may tend to purchase after some examinations, which also explains that as ℓ increases, $\#(\ell)E2P$ also increases. Users in the Rec15 dataset like to take consecutive purchases, so $\#(\ell)P2P$ is far more than $\#(\ell)E2P$ and $\#(\ell)P2E$, and $\#(\ell)P2E$ is quite small. In the other datasets, users prefer to make a single purchase and then turn to examine other items, so $\#(\ell)E2P$ and $\#(\ell)P2E$ are close, and $\#(\ell)P2P$ is the smallest.

We then take the dataset Rec15 as an example to analyze the relationships among the behavior transition vectors. For visualization, we use principal component analysis [23] to reduce the high-dimensionality behavior transition vectors to 2D vectors. The averaged translation vectors of all the users

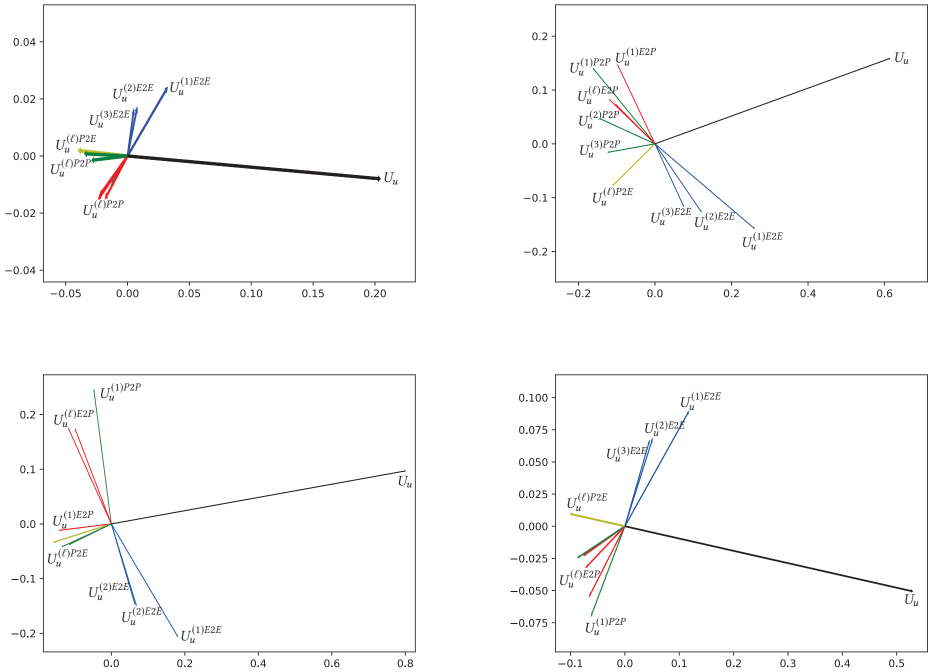

Fig. 5 Visualization of the behavior transition vectors, including the averaged vectors of all users (left, top), the vectors of the user with the longest interaction sequence (right, top), the vectors of the user with a medium length interaction sequence (left, bottom) and the vectors of the user with a short length interaction sequence (right, bottom). Notice that we omit the subscript for simplicity.

and the translation vectors of the users with the longest interaction sequence, a medium length sequence and a short length sequence are shown in Fig. 5.

From Fig. 5, we can have the following observations: (i) The global translation vector $U_{u\cdot}$ that captures the users' global preference is much longer than the behavior transition vectors, where the latter rectifies the former by exploiting the behavior transition information. (ii) The behavior transition vectors of the same category (of E2E, E2P, P2E or P2P) are close to each other, which means that these vectors are effective in capturing the users' behavior transitions. (iii) The behavior transition vectors of a lower order (e.g., $\ell = 1$) are usually longer than the ones of a higher order (e.g., $\ell = 3$), meaning that a near item has a stronger impact on a target item, which showcases the effectiveness of considering L preceding items in our TransRec++.

5.5.5 Analyses of Different Data Input for Sequential Recommendation Methods

For each sequential recommendation method, we have two variants in which we exploit the purchase feedback only (denoted as $\mathcal{P}$) and both the purchase feedback and the examination feedback (denoted as $\mathcal{P} \cup \mathcal{E}$), respectively. We report their recommendation performance in Fig. 6.

From Fig. 6, we can see that for all the three methods (i.e., FPMC, Fossil and TransRec) and all the datasets except ML1M, the variant with $\mathcal{P} \cup \mathcal{E}$ performs better than the corresponding variant with $\mathcal{P}$, which is consistent with our intuition that examinations and purchases are usually complementary in learning users' true preferences. Since ML1M is a simulated dataset from movie ratings rather than a real dataset with examinations and purchases, the simulated examination feedback may have an adverse effect on users' preference learning.

5.5.6 Extension to Cases with More Types of Behaviors

In this section, we study the performance of our TransRec++ when more than two types of behaviors are taken into consideration. For the UB and Tmall datasets, apart from keeping four types of behaviors, i.e., examination, purchase, add-to-favorite and add-to-cart, we preprocess them in the same way as that in Section 5.1. As we have discussed in Section 4.6, when the behaviors add-to-favorite and add-to-cart are involved, we can generalize our TransRec++ by includ-

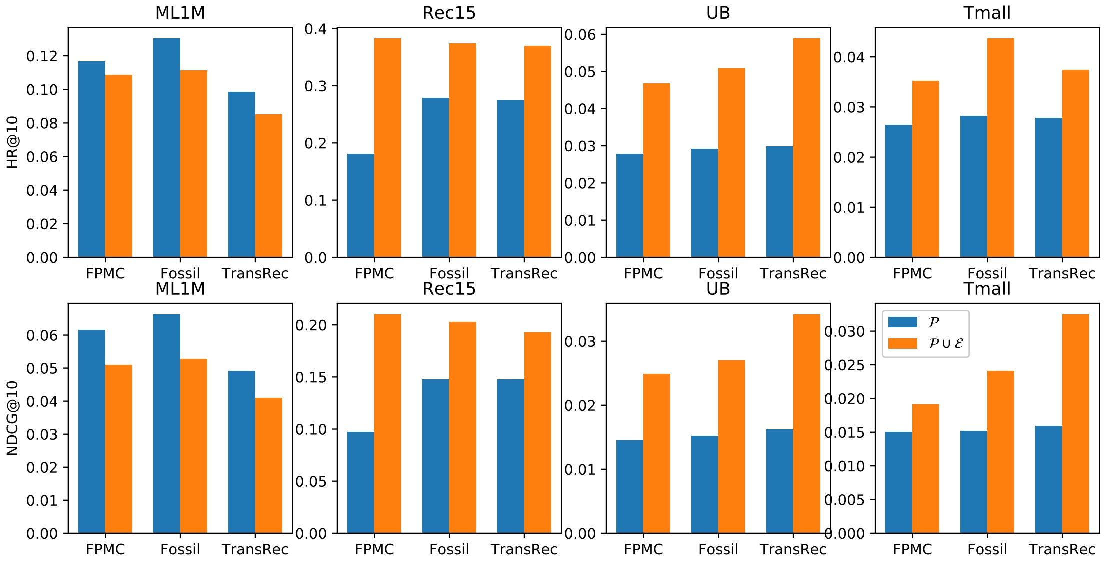

Fig. 6 Recommendation performance of sequential recommendation methods with purchases (i.e., $\mathcal{P}$) and both purchases and examinations (i.e., $\mathcal{P} \cup \mathcal{E}$) on the four datasets.

ing some additional behavior transition vectors. We report the results of our TransRec++ with two types of behaviors (#*bh* = 2) and four types of behaviors (#*bh* = 4) in Table 6, where the best results are marked in bold.

Table 6 Performance of TransRec++ with more types of behaviors.

L	#*bh*	UB		Tmall	
		HR@10	NDCG@10	HR@10	NDCG@10
1	2	0.0661	0.0413	0.0540	0.0387
	4	**0.0729**	**0.0466**	**0.0546**	**0.0401**
2	2	0.0647	0.0390	0.0593	0.0377
	4	**0.0760**	**0.0451**	**0.0614**	**0.0385**
3	2	0.0627	0.0355	0.0545	0.0324
	4	**0.0726**	**0.0429**	**0.0570**	**0.0337**

From Table 6, we can see that our TransRec++ achieves better performance when more types of behaviors are considered, which showcases that our TransRec++ can deal with multiple types of behaviors effectively.

5.5.7 Application to Deep Learning-Based Methods

In this section, we apply our behavior transition vectors to two deep learning-based methods (i.e., RIB [27] and BINN [8]). RIB and BINN use an RNN to model the relationships between different behaviors implicitly, but do not explicitly consider the behavior transitions between different types of behaviors. In order to verify whether the behavior transition vectors can be applied to deep learning methods, we integrate the modeling of E2E, E2P, P2E and P2P into RIB and BINN. Specifically, we add the corresponding behavior transition vector from the current behavior to the next behavior at each time step in RIB and BINN. The results on HR@10 are shown in Fig. 7, from which we can see that for all the datasets, both the enhanced methods (i.e., RIB++ and BINN++) clearly surpass the corresponding methods (i.e., RIB and BINN) in most cases, which shows that the proposed behavior transition vectors are of good generality and helpful in improving the two deep learning-based methods.

6 Conclusions and Future Work

In this paper, we study sequential recommendation with heterogeneous feedback (SRHF) and propose a novel and generic translation-based model, i.e., TransRec++, which is the first shallow learning-based method for the studied problem as far as we know. Our model introduces a series of high-order user behavior transition vectors to capture the relationships among the heterogeneous sequential feedback, and thus exploits the heterogeneous feedback and sequential information in an intuitive and unified way, which empowers our TransRec++ with good interpretability. We conduct extensive experiments on four datasets, and show that our TransRec++ is able to outperform several state-of-the-art meth-

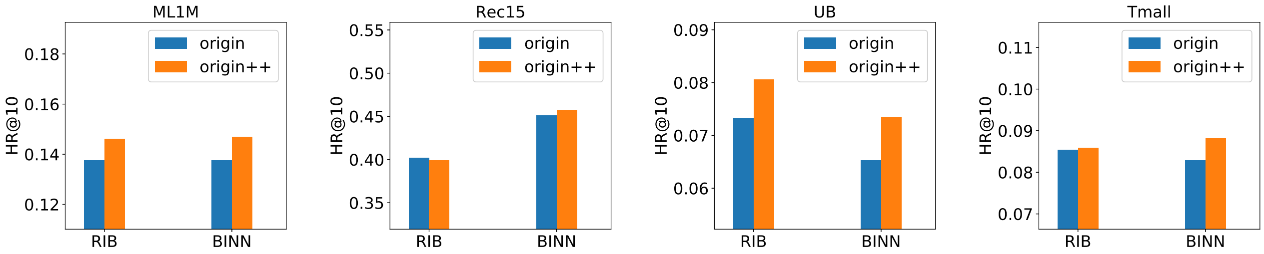

Fig. 7 Recommendation performance of deep learning-based methods (i.e., RIB and BINN, denoted by "origin") and their enhanced versions with behavior transition vectors (i.e., RIB++ and BINN++, denoted by "origin++") on the four datasets.

ods significantly in most cases. Furthermore, we analyze the relationships among the translation vectors, including users' global translation vectors and the transition vectors of different orders and different kinds of behavior transitions, which gives us more insights on TransRec++. We also study our TransRec++ in recommendation problems with four different types of behaviors, and apply the proposed behavior transition vectors to two deep learning-based methods, both of which achieve very promising results showcasing the flexibility and generality of our solution for SRHF.

For future works, we are interested in studying the effectiveness of the proposed behavior transition vectors in more deep learning paradigms [25], and generalizing our TransRec++ to include more information such as knowledge graph [22] and review text [26]. Moreover, we are also interested in designing a federated version [11, 24] of our TransRec++ for privacy protection when utilizing users' heterogeneous sequential feedback.

Acknowledgements We thank the support of National Natural Science Foundation of China Nos. 62172283 and 61836005.

References

1. Oren Barkan and Noam Koenigstein. ITEM2VEC: Neural item embedding for collaborative filtering. In *Proceedings of the 26th IEEE International Workshop on Machine Learning for Signal Processing*, MLSP'16, pages 1–6, 2016.
2. Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems*, 39(1):1–42, 2020.
3. F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015.
4. Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys'17, pages 161–169, 2017.
5. Ruining He and Julian McAuley. Fusing similarity models with Markov chains for sparse sequential recommendation. In *Proceedings of 16th International Conference on Data Mining*, ICDM'16, pages 191–200, 2016.
6. Santosh Kabbur, Xia Ning, and George Karypis. FISM: Factored item similarity models for top-N recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'13, pages 659–667, 2013.
7. Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'08, pages 426–434, 2008.
8. Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'18, pages 1734–1743, 2018.
9. Feng Liang, Wei Dai, Yunfeng Huang, Weike Pan, and Zhong Ming. PAT: Preference-aware transfer learning for recommendation with heterogeneous feedback. In *Proceedings of 2020 International Joint Conference on Neural Networks*, IJCNN'20, pages 1–7, 2020.
10. Wenjing Meng, Deqing Yang, and Yanghua Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'20, pages 1091–1100, 2020.
11. Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. FedFast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'20, pages 1234–1242, 2020.
12. Weike Pan, Mengsi Liu, and Zhong Ming. Transfer learning for heterogeneous one-class collaborative filtering. *IEEE Intelligent Systems*, 31(4):43–49, 2016.
13. Weike Pan and Zhong Ming. Collaborative recommendation with multiclass preference context. *IEEE Intelligent Systems*, 32(2):45–51, 2017.
14. Weike Pan, Shanchuan Xia, Zhuode Liu, Xiaogang Peng, and Zhong Ming. Mixed factorization for collaborative recommendation with

heterogeneous explicit feedbacks. *Information Sciences*, 332:84–93, 2016.

15. Weike Pan, Qiang Yang, Wanling Cai, Yaofeng Chen, Qing Zhang, Xiaogang Peng, and Zhong Ming. Transfer to rank for heterogeneous one-class collaborative filtering. *ACM Transactions on Information Systems*, 37(1):10:1–10:20, 2019.
16. Weike Pan, Hao Zhong, Congfu Xu, and Zhong Ming. Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks. *Knowledge-Based Systems*, 73:173–180, 2015.
17. Arkadiusz Paterek. Improving regularized Singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.
18. Xiaogang Peng, Yaofeng Chen, Yuchao Duan, Weike Pan, and Zhong Ming. RBPR: Role-based bayesian personalized ranking for heterogeneous one-class collaborative filtering. In *CEUR Workshop Proceedings of Late-breaking Results, Posters, Demos, Doctoral Consortium and Workshops Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalisation (UMAP 2016)*, volume 1618, 2016.
19. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI'09, pages 452–461, 2009.
20. Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW'10, pages 811–820, 2010.
21. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, pages 285–295, 2001.
22. Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
23. Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.
24. Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):12:1–12:19, 2019.
25. Zhiqian Zhang, Chenliang Li, Zhiyong Wu, Aixin Sun, Dengpan Ye, and Xiangyang Luo. NEXT: A neural network framework for next POI recommendation. *Frontiers of Computer Science*, 14(2):314–333, 2020.
26. Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, WSDM'17, pages 425–434, 2017.
27. Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, WSDM'18, pages 727–735, 2018.