Adaptive Transfer Learning for Heterogeneous One-Class Collaborative Filtering

Xiancong Chen, Weike Pan*, Zhong Ming*
National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
chenxiancong@email.szu.edu.cn, {panweike,mingz}@szu.edu.cn

***Abstract*—In this paper, we study a recent and important recommendation problem called heterogeneous one-class collaborative filtering (HOCCF), where we have two different types of one-class feedback, i.e., a set of browses and a set of purchases, for preference learning. Previous methods exploit the browses and purchases by extending some existing methods on modeling homogeneous one-class feedback in an integrative or sequential manner. However, an integrative method may be of high complexity in model training due to the expanded prediction rule, and a sequential method is of inconvenience in deployment because more than one parametric models have to be maintained. In this paper, we convert the HOCCF problem to an adaptive transfer learning task, where we first model browses via a factorization model from a perspective of the role of browser, and then adaptively refine the learned model parameters via purchases from a perspective of the dependent role of purchaser. Based on this conversion, we design a novel solution called role-based adaptive factorization (ROAF), and then derive two specific variants with pairwise preference learning and pointwise preference learning. Finally, we conduct extensive empirical studies on two large datasets, and find that our ROAF is a very promising solution in terms of recommendation accuracy, besides its convenience of one single parametric model in deployment.**

***Index Terms*—Transfer Learning, Heterogeneous One-Class Collaborative Filtering, Adaptive Factorization**

I. Introduction

Recommender systems play an increasingly important role in helping users find products that meet their need or interest. Based on the observation that users' preferences can be reflected by their heterogeneous types of feedback towards items, it is intuitive and valuable to exploit these feedback for better recommendation performance. In general, users' feedback can be classified as explicit feedback such as ratings, and implicit feedback (or one-class feedback) such as browses and purchases. However, the major shortcoming of explicit feedback is that they are very few and are not always available. Therefore, most state-of-the-art methods focus on modeling implicit feedback. OCCF [1] is proposed to model one single type of one-class feedback such as purchases, but the methods of OCCF may suffer the mediocre recommendation performance due to the sparsity problem.

To alleviate the above problems, heterogeneous one-class collaborative filtering is proposed in recent years, in which more than one single type of one-class feedback are available, e.g., purchases and browses. Our goal is to exploit the connection and difference among these heterogeneous feedback. Specifically, the connection is that users will browse lots of products before they make a decision to purchase them. The difference is that purchases are fewer but have a stronger indication of users' preferences while browses are more abundant but have a weaker indication.

Some recent methods are proposed to solve the HOCCF problem. In an integrative manner, adaptive Bayesian personalize ranking (ABPR) [2] learns a confidence for each browse, and transfer via joint similarity learning (TJSL) [3] learns a joint similarity of two different types of one-class feedback. However, these two methods are limited by their inefficiency due to the calculation of the confidence and the expanded prediction rule that contains additional terms. View-enhanced eALS (VALS) [4] adopts a fast learning algorithm but is associated with too many hyper-parameters to be tuned easily. In a sequential manner, role-based Bayesian personalize ranking (RBPR) [5] is a two-stage method from a perspective of users' roles. But this solution is lack of principled foundation for preference learning, and is also of inconvenience in deployment because of one candidate list of items to be maintained. Role-based transfer to rank (RoToR) [6], which contains two specific variants in an integrative manner and a sequential manner, respectively, also suffers from the shortcoming of inefficiency in training and inconvenience in deployment due to the expanded prediction rules and two (instead of one single) parametric models to be maintained.

As a response, we design a simple but effective transfer learning solution termed as role-based adaptive factorization (ROAF), which contains a pairwise preference learning variant, i.e., ROAF(pai.), and a pointwise preference learning variant, i.e., ROAF(poi.). Specifically, in the browser stage, we model the role of browser and optimize the parameters using both purchasing and browsing behaviors. Then in the purchaser stage, we transfer the coarse-grained parameters from the browser stage and refine them adaptively using the purchasing behaviors only from the perspective of a purchaser. In both stages, we adopt short and simple prediction rules, and merely need to maintain one single parametric model. Our solution thus avoids the inefficiency of the expanded prediction rule

*: Corresponding author.

and the inconvenience of the parametric models and candidate list of items. Experimental results on two large datasets show the effectiveness of our ROAF in terms of recommendation accuracy as well as training and test efficiency in comparison with the competitive baseline methods.

II. Related Work

Recommendation with heterogeneous one-class feedback or HOCCF has recently received more attention in the community of recommender systems because it provides us with tremendous opportunities to learn users' preferences better by leveraging the abundant browses besides the purchases. Some methods have thus been proposed to deal with the HOCCF problem, which are categorized into factorization-based methods and transfer learning-based methods, respectively.

A. Factorization in Collaborative Filtering

In OCCF, the well-known classical factorization-based method, i.e., matrix factorization (MF) [7], can capture the interactions between users and items with their latent feature vectors, which is able to alleviate the sparsity problem to some extent. Hence, many researchers adopt this form to develop their solutions, among which Bayesian personalized ranking (BPR) [8] is a seminal one. Recently, there are several new factorization-based solutions for the HOCCF problem. For example, Bayesian personalized ranking for heterogeneous implicit feedback (BPRH) [9] extends the BPR framework, and calculates the coefficient for each user to quantify the correlation between the purchasing behaviors and other behaviors. But this method is not easy to be generalized due to the specially designed calculation of the coefficient and the sampling strategy. Very recently, VALS [4] uses three different terms in the objective function to model users' preferences towards the purchased items, browsed items and un-browsed items, achieving higher accuracy. However, the complex loss function contains too many hyper-parameters, which have to be tuned from large tuning ranges, limiting its applicability. Behavior-intensive neural network (BINN) [10] and the recommendation framework for modeling micro behaviors (RIB) [11] are two deep learning based methods for HOCCF with additional sequential information, in which they first concatenate the behavior embedding with the corresponding item embedding and then feed them to a recurrent neural network.

B. Transfer Learning in Collaborative Filtering

Transfer learning [12] is proposed to transfer knowledge from an auxiliary data for the purpose of assisting the learning task on a target data. Some recent works are proposed to solve the HOCCF problem via a different way of knowledge transfer. From the perspective of integrative transfer learning, ABPR [2] and TJSL [3] are proposed to alleviate the uncertainty of the users' preferences using both the purchasing behaviors and the browsing behaviors. ABPR is able to learn a confidence weight for each browsing record and then integrates it to the objective function of the target learning task. TJSL integrates the similarity between the purchased items and browsed items into a factorized model. RoToR(int.) [6] integrates the neighborhood-based method and factorization-based method into one single prediction rule. However, these methods may suffer from long training time due to the expanded prediction rule. RoToR(seq.) [6] and RBPR [5] are two recently proposed methods in a sequential transfer learning manner. Specifically, RoToR(seq.) combines a neighborhood-based method and a factorization-based method sequentially. And RBPR is the most closest work to ours, which is also a two-stage learning method. The main difference is that our solution makes a refinement of the model parameters adaptively while RBPR re-ranks the candidate lists obtained from the first stage. Both RoToR(seq.) and RBPR need to maintain more than one parametric models as well as the candidate lists served as the input data in the second stage, making them inconvenient in deployment. There are also some cross-domain transfer learning methods [13], [14], but we mainly focus on one specific domain with heterogeneous feedback in this paper.

III. Heterogeneous One-Class Collaborative Filtering

A. Problem Definition

In the studied HOCCF problem, we have some feedback from n users and m items, including some relatively positive feedback such as purchases $\mathcal{R}^{\mathcal{P}} = \{(u,i)\}$ and some implicit feedback such as browses $\mathcal{R}^{\mathcal{B}} = \{(u,i)\}$. For notation simplicity and without loss of generality, we assume $\mathcal{R}^{\mathcal{P}} \cap \mathcal{R}^{\mathcal{B}} = \emptyset$, i.e., if an item i is first browsed and then purchased by a user u, we will keep the (u,i) pair in $\mathcal{R}^{\mathcal{P}}$ only. As for a typical user $u \in \mathcal{U}$, we have a set of purchased items, i.e., $\mathcal{I}_u^{\mathcal{P}}$, and a set of browsed items, i.e., $\mathcal{I}_u^{\mathcal{B}}$, where $\mathcal{I}_u^{\mathcal{P}} \cap \mathcal{I}_u^{\mathcal{B}} = \emptyset$. Our goal is then to exploit these two types of one-class feedback and recommend a ranked list of items from the un-purchased items $\mathcal{I} \backslash \mathcal{I}_u^{\mathcal{P}}$ for each user u. We list some commonly used notations and their explanations in Table I.

B. Challenges

In order to model the heterogeneous one-class feedback, some very recent works focus on integrative and sequential approaches [3], [5], [6], where the former exploits the different types of feedback via an expanded and complex prediction rule and the latter achieves this via two dependent stages with candidate lists of items served as the shared knowledge. Though these two methods have achieved the state-of-the-art performance in terms of recommendation accuracy, the main limitation is their high complexity. Specifically, the expanded and complex prediction rule in the integrative method will cause inefficiency in training due to the additional parameters to be learned, and the candidate lists of items as well as two different parametric models in the sequential method will cause inconvenience in deployment and maintenance. We include a detailed and quantitative study and discussion for this issue in our empirical studies.

TABLE I
SOME NOTATIONS AND EXPLANATIONS.

Notation	Explanation
n	number of users
m	number of items
$u \in \{1, 2, \ldots, n\}$	user ID
$i \in \{1, 2, \ldots, m\}$	item ID
$\mathcal{U} = \{u\}$, $\lvert\mathcal{U}\rvert = n$	the whole set of users
$\mathcal{I} = \{i\}$, $\lvert\mathcal{I}\rvert = m$	the whole set of items
$\mathcal{R}^{\mathcal{P}} = \{(u, i)\}$	the whole set of purchases
$\mathcal{R}^{\mathcal{B}} = \{(u, i)\}$	the whole set of browses
$\mathcal{R}^{\mathcal{A}} = \{(u, i)\}$	the whole set of absent pairs
$\mathcal{R} = \mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{B}} \cup \mathcal{R}^{\mathcal{A}}$	the whole set of pairs
$\mathcal{I}_u^{\mathcal{P}} = \{i \vert (u, i) \in \mathcal{R}^{\mathcal{P}}\}$	purchased items w.r.t. u
$\mathcal{I}_u^{\mathcal{B}} = \{i \vert (u, i) \in \mathcal{R}^{\mathcal{B}}\}$	browsed items w.r.t. u
$\mathcal{I}_u^{\mathcal{A}} = \{i \vert (u, i) \in \mathcal{R}^{\mathcal{A}}\}$	absent items w.r.t u
$\mathcal{R}^{\overline{\mathcal{B} \cup \mathcal{P}}} = \{(u, i)\} \subseteq \mathcal{R}^{\mathcal{A}}$	sampled absent pairs
$\mathcal{R}^{\overline{\mathcal{P}}} = \{(u, i)\} \subseteq \mathcal{R} \backslash \mathcal{R}^{\mathcal{P}}$	sampled non-purchases
d	number of latent dimension
$U_{u\cdot}, \tilde{U}_{u\cdot} \in \mathcal{R}^{1 \times d}$	user u's feature vector
$V_{i\cdot}, \tilde{V}_{i\cdot} \in \mathcal{R}^{1 \times d}$	item i's feature vector
$b_u, \tilde{b}_u \in \mathcal{R}$	user bias
$b_i, \tilde{b}_i \in \mathcal{R}$	item bias
$\hat{r}_{ui}, \tilde{\hat{r}}_{ui}$	prediction w.r.t. u and i
γ	learning rate
T	iteration number

C. Overall of Our Solution

In order to address the challenge, we propose a simple but effective solution via adaptive knowledge transfer from a browser stage to a purchaser stage. Specifically, in the browser stage, we optimize the initialized parameters using both purchasing and browsing behaviors. Then in the purchaser stage, we transfer the coarse-grained parameters from the browser stage and refine them adaptively using the purchasing behaviors only. In both stages, we adopt a compact prediction rule defined on one single type of behaviors instead of an expanded one on two types of behaviors. Besides, we adaptively refine the model parameters rather than re-rank the candidate lists. We illustrate it in Figure 1.

IV. OUR SOLUTION

In this section, we will first introduce the main idea of our proposed adaptive transfer learning solution, and then describe two specific variants, i.e., pairwise preference learning and pointwise preference learning.

A. Role-based Adaptive Factorization

In our ROAF, we view the two types of one-class feedback from a perspective of users' roles, i.e., browser and purchaser for the browses and purchases, respectively. Based on this perspective, we design our solution in an adaptive manner. Firstly, we focus on the role of browser, and train a model for the purpose of identifying some likely to be browsed items. Secondly, we adaptively refine the learned model parameters via purchases only aiming to find some items that will affect users' final purchase decisions. Technically, in order to refine the parameters of our factorization-based model, we adopt

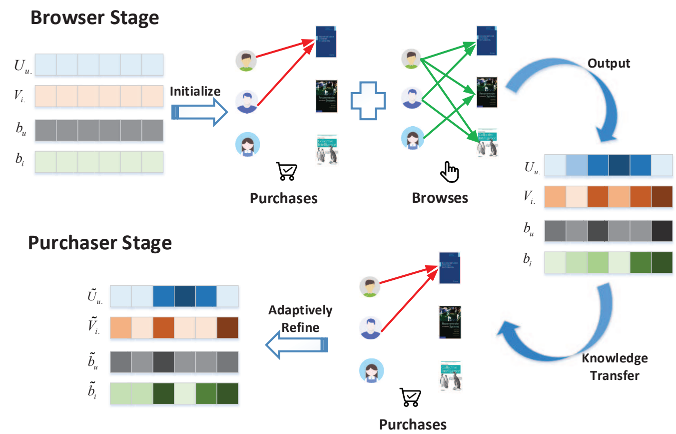

Fig. 1. Illustration of our proposed role-based adaptive factorization (ROAF) for heterogeneous one-class collaborative filtering (HOCCF), including the browser stage and the purchaser stage.

biased regularization in the second stage so that the learned model parameters can be mildly adapted from the browser stage to the purchaser stage. For this reason, we name our solution as role-based adaptive factorization (ROAF).

As for user behavior modeling or user preference learning in either the browser stage or the purchaser stage, we develop two specific variants, including ROAF via pairwise preference learning and ROAF via pointwise preference learning, which will be described in detail in the sequel.

B. Pairwise Preference Learning

In our first variant of ROAF, i.e., pairwise preference learning, we follow the seminal work BPR [8] and assume that a user prefers an interacted (i.e., browsed or purchased) item to an un-interacted item.

1) Pairwise Preference Learning in Browser Stage: In the browser stage, we first combine the set of browsed items and the set of purchased items w.r.t. a certain user u, i.e., $\mathcal{I}_u^{\mathcal{P}} \cup \mathcal{I}_u^{\mathcal{B}}$, for the purpose of obtaining an augmented set of interacted items, and then obtain an overall objective function,

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u^{\mathcal{P}} \cup \mathcal{I}_u^{\mathcal{B}}} \sum_{j \in \mathcal{I}_u^{\mathcal{A}}} -\log \sigma(\hat{r}_{ui} - \hat{r}_{uj}) + \text{reg}(\Theta), \quad (1)$$

where $\Theta = \{U_{u\cdot}, V_{i\cdot}, b_i \vert u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the model parameters to be learned, $\sigma(\hat{r}_{ui} - \hat{r}_{uj}) = 1/(1+\exp^{-(\hat{r}_{ui} - \hat{r}_{uj})})$ is the sigmoid function used to approximate the probability of $\hat{r}_{ui} > \hat{r}_{uj}$ according to the pairwise preference assumption, and the regularization term $\text{reg}(\Theta) = \frac{\alpha_u}{2}\lvert\lvert U_{u\cdot}\rvert\rvert^2 + \frac{\alpha_v}{2}\lvert\lvert V_{i\cdot}\rvert\rvert^2 + \frac{\alpha_v}{2}\lvert\lvert V_{j\cdot}\rvert\rvert^2 + \frac{\beta_v}{2}\lvert\lvert b_i\rvert\rvert^2 + \frac{\beta_v}{2}\lvert\lvert b_j\rvert\rvert^2$ is used to avoid overfitting during parameter learning. Notice that the prediction rule for the preference of user u to item i is $\hat{r}_{ui} = U_{u\cdot}V_{i\cdot}^T + b_i$.

In order to solve the optimization problem in Eq.(1), we calculate the gradients of the model parameters for a triple

(u, i, j) as follows,

$$\nabla U_{u\cdot} = -\sigma(-(\hat{r}_{ui} - \hat{r}_{uj}))(V_{i\cdot} - V_{j\cdot}) + \alpha_u U_{u\cdot},$$
$$\nabla V_{i\cdot} = -\sigma(-(\hat{r}_{ui} - \hat{r}_{uj}))U_{u\cdot} + \alpha_v V_{i\cdot},$$
$$\nabla V_{j\cdot} = -\sigma(-(\hat{r}_{ui} - \hat{r}_{uj}))(-U_{u\cdot}) + \alpha_v V_{j\cdot},$$
$$\nabla b_i = -\sigma(-(\hat{r}_{ui} - \hat{r}_{uj})) + \beta_v b_i,$$
$$\nabla b_j = -\sigma(-(\hat{r}_{ui} - \hat{r}_{uj}))(-1) + \beta_v b_j.$$

With the above gradients, we can then update the model parameters in a commonly used stochastic gradient descent (SGD) algorithm as follows,

$$\theta_{\tau+1} \leftarrow \theta_\tau - \gamma \nabla \theta_\tau,$$

where γ is the learning rate.

2) Pairwise Model Adaptation in Purchaser Stage: In the purchaser stage, we make use of purchases only to refine the model parameters learned in the browser stage, i.e., Θ. The objective function is similar to that of the first stage,

$$\min_{\tilde{\Theta}} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u^{\mathcal{P}}} \sum_{j \in \mathcal{I} \backslash \mathcal{I}_u^{\mathcal{P}}} -\log \sigma(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj}) + \text{reg}(\tilde{\Theta}|\Theta), \quad (2)$$

where $\tilde{\Theta} = \{\tilde{U}_{u\cdot}, \tilde{V}_{i\cdot}, \tilde{b}_i | u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the parameters to be learned, $\text{reg}(\tilde{\Theta}|\Theta) = \frac{\alpha_u}{2}||\tilde{U}_{u\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{i\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{j\cdot}||^2 + \frac{\beta_v}{2}||\tilde{b}_i||^2 + \frac{\beta_v}{2}||\tilde{b}_j||^2 + \frac{\alpha_u}{2}||\tilde{U}_{u\cdot} - U_{u\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{i\cdot} - V_{i\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{j\cdot} - V_{j\cdot}||^2 + \frac{\beta_v}{2}||\tilde{b}_i - b_i||^2 + \frac{\beta_v}{2}||\tilde{b}_j - b_j||^2$ is the biased regularization term used to make the new model parameters $\tilde{\Theta}$ to be not very different from Θ because of the close relationship of the two stages. Notice that the prediction rule is similar to that in the browser stage, i.e., $\tilde{\hat{r}}_{ui} = \tilde{U}_{u\cdot}\tilde{V}_{i\cdot} + \tilde{b}_i$.

We then calculate the gradients again for a triple (u, i, j),

$$\nabla \tilde{U}_{u\cdot} = -\sigma(-(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj}))(\tilde{V}_{i\cdot} - \tilde{V}_{j\cdot}) + \alpha_u \tilde{U}_{u\cdot} + \alpha_u(\tilde{U}_{u\cdot} - U_{u\cdot}),$$
$$\nabla \tilde{V}_{i\cdot} = -\sigma(-(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj}))\tilde{U}_{u\cdot} + \alpha_v \tilde{V}_{i\cdot} + \alpha_v(\tilde{V}_{i\cdot} - V_{i\cdot}),$$
$$\nabla \tilde{V}_{j\cdot} = -\sigma(-(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj}))(-\tilde{U}_{u\cdot}) + \alpha_v \tilde{V}_{j\cdot} + \alpha_v(\tilde{V}_{j\cdot} - V_{j\cdot}),$$
$$\nabla \tilde{b}_i = -\sigma(-(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj})) + \beta_v \tilde{b}_i + \beta_v(\tilde{b}_i - b_i),$$
$$\nabla \tilde{b}_j = -\sigma(-(\tilde{\hat{r}}_{ui} - \tilde{\hat{r}}_{uj}))(-1) + \beta_v \tilde{b}_j + \beta_v(\tilde{b}_j - b_j),$$

and update the model parameters in the SGD algorithm,

$$\tilde{\theta}_{\tau+1} \leftarrow \tilde{\theta}_\tau - \gamma \nabla \tilde{\theta}_\tau.$$

We depict the whole pairwise preference learning algorithm of ROAF, i.e., ROAF(pai.), in Algorithm 1. Specifically, in the browser stage, we use both the browses and purchases to learn a basic model Θ; and in the purchaser stage, we adopt a warm-start strategy to train the model $\tilde{\Theta}$, i.e., initialization via $\tilde{\Theta} = \Theta$, and adapt the model parameters Θ mildly.

C. Pointwise Preference Learning

In our second variant of ROAF, i.e., pointwise preference learning, we follow a recent work MF(LogisticLoss) [15] and assume that a user likes an interacted (i.e., browsed or purchased) item and dislikes an un-interacted one.

Algorithm 1 The algorithm of ROAF(pai.).

```
// The browser stage
Input: R^B, R^P.
Output: Θ.
Initialization: Initialize model parameters Θ.
for t = 1, ..., T do
    for t_2 = 1, ..., |R^P ∪ R^B| do
        Randomly pick up a pair (u, i) from R^P ∪ R^B.
        Randomly pick up an item j from I_u^A.
        Calculate the gradients.
        Update the corresponding model parameters.
    end for
end for
// The purchaser stage
Input: R^P, Θ.
Output: Θ̃.
Initialization: Initialize model parameters Θ̃ = Θ.
for t = 1, ..., T do
    for t_2 = 1, ..., |R^P| do
        Randomly pick up a pair (u, i) from R^P.
        Randomly pick up an item j from I\I_u^P.
        Calculate the gradients.
        Update the corresponding model parameters.
    end for
end for
```

1) Pointwise Preference Learning in Browser Stage: In the browser stage, we represent the probability that a user u browses an item i as $\sigma(\hat{r}_{ui})$, where $\hat{r}_{ui} = U_{u\cdot}V_{i\cdot}^T + b_u + b_i$. Due to the lack of un-interacted (i.e., un-browsed) items, we randomly sample some absent pairs $\mathcal{R}^{\overline{\mathcal{B} \cup \mathcal{P}}}$ from $\mathcal{R} \backslash (\mathcal{R}^{\mathcal{B}} \cup \mathcal{R}^{\mathcal{P}})$ [16]. We then reach the objective function,

$$\min_{\Phi} \sum_{(u,i) \in \mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{B}} \cup \mathcal{R}^{\overline{\mathcal{B} \cup \mathcal{P}}}} -\log(1 + \exp(-r_{ui}\hat{r}_{ui})) + \text{reg}(\Phi), \quad (3)$$

where $\Phi = \{U_{u\cdot}, V_{i\cdot}, b_u, b_i | u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the parameters to be learned, and the regularization term $\text{reg}(\Phi) = \frac{\alpha_u}{2}||U_{u\cdot}||^2 + \frac{\alpha_v}{2}||V_{i\cdot}||^2 + \frac{\beta_u}{2}||b_u||^2 + \frac{\beta_v}{2}||b_i||^2$ is used to avoid overfitting. Notice that the label $r_{ui} = 1$ if $(u, i) \in \mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{B}}$ and $r_{ui} = -1$ if $(u, i) \in \mathcal{R}^{\overline{\mathcal{B} \cup \mathcal{P}}}$, which means to maximize and minimize the browsing likelihood, respectively.

Similarly, we can update the model parameters using the following gradients w.r.t. a (u, i) pair in the SGD algorithm.

$$\phi_{\tau+1} \leftarrow \phi_\tau - \gamma \nabla \phi_\tau,$$

where the gradients of the model parameters are as follows,

$$\nabla U_{u\cdot} = -\frac{r_{ui}}{1 + \exp(r_{ui}\hat{r}_{ui})} V_{i\cdot} + \alpha_u U_{u\cdot},$$
$$\nabla V_{i\cdot} = -\frac{r_{ui}}{1 + \exp(r_{ui}\hat{r}_{ui})} U_{u\cdot} + \alpha_v V_{i\cdot},$$
$$\nabla b_u = -\frac{r_{ui}}{1 + \exp(r_{ui}\hat{r}_{ui})} + \beta_u b_u,$$
$$\nabla b_i = -\frac{r_{ui}}{1 + \exp(r_{ui}\hat{r}_{ui})} + \beta_v b_i.$$

2) Pointwise Model Adaptation in Purchaser Stage: In the purchaser state, we have a similar objective function,

$$\min_{\tilde{\Phi}} \sum_{(u,i)\in\mathcal{R}^{\mathcal{P}}\cup\mathcal{R}^{\overline{\mathcal{P}}}} -\log(1+\exp(-r_{ui}\hat{\tilde{r}}_{ui})) + \text{reg}(\tilde{\Phi}|\Phi), \quad (4)$$

where $\tilde{\Phi} = \{\tilde{U}_{u\cdot}, \tilde{V}_{i\cdot}, \tilde{b}_u, \tilde{b}_i | u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the parameters to be learned, $\text{reg}(\tilde{\Phi}|\Phi) = \frac{\alpha_u}{2}||\tilde{U}_{u\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{i\cdot}||^2 + \frac{\beta_u}{2}||\tilde{b}_u||^2 + \frac{\beta_v}{2}||\tilde{b}_i||^2 + \frac{\alpha_u}{2}||\tilde{U}_{u\cdot} - U_{u\cdot}||^2 + \frac{\alpha_v}{2}||\tilde{V}_{i\cdot} - V_{i\cdot}||^2 + \frac{\beta_u}{2}||\tilde{b}_u - b_u||^2 + \frac{\beta_v}{2}||\tilde{b}_i - b_i||^2$ is again the biased regularization for model adaptation. Notice that $\hat{\tilde{r}}_{ui} = \tilde{U}_{u\cdot}\tilde{V}_{i\cdot}^T + \tilde{b}_u + \tilde{b}_i$ is the prediction rule, $\mathcal{R}^{\overline{\mathcal{P}}}$ is sampled from $\mathcal{R}\backslash\mathcal{R}^{\mathcal{P}}$, and $r_{ui} = 1$ if $(u,i) \in \mathcal{R}^{\mathcal{P}}$ and $r_{ui} = -1$ otherwise.

Similarly, we can update the model parameters w.r.t. to a randomly sampled (u,i) pair,

$$\tilde{\phi}_{\tau+1} \leftarrow \tilde{\phi}_{\tau} - \gamma\nabla\tilde{\phi}_{\tau},$$

where the gradients of the model parameters are as follows,

$$\begin{aligned}
\nabla\tilde{U}_{u\cdot} &= -\frac{r_{ui}}{1+\exp(r_{ui}\hat{\tilde{r}}_{ui})}\tilde{V}_{i\cdot} + \alpha_u\tilde{U}_{u\cdot} + \alpha_u(\tilde{U}_{u\cdot} - U_{u\cdot}),\\
\nabla\tilde{V}_{i\cdot} &= -\frac{r_{ui}}{1+\exp(r_{ui}\hat{\tilde{r}}_{ui})}\tilde{U}_{u\cdot} + \alpha_v\tilde{V}_{i\cdot} + \alpha_v(\tilde{V}_{i\cdot} - V_{i\cdot}),\\
\nabla\tilde{b}_u &= -\frac{r_{ui}}{1+\exp(r_{ui}\hat{\tilde{r}}_{ui})} + \beta_u\tilde{b}_u + \beta_u(\tilde{b}_u - b_u),\\
\nabla\tilde{b}_i &= -\frac{r_{ui}}{1+\exp(r_{ui}\hat{\tilde{r}}_{ui})} + \beta_v\tilde{b}_i + \beta_v(\tilde{b}_i - b_i),
\end{aligned}$$

We depict the pointwise learning algorithm of ROAF, i.e., ROAF(poi.), in Algorithm 2, which is similar to that in Algorithm 1. The main difference is that we have to randomly sample a set of absent pairs in the browser stage and a set of un-interacted pairs (i.e., non-purchases) in the purchaser stage due to the lack of negative feedback in pointwise learning.

D. Discussions

We can see that either ROAF(pai.) or ROAF(poi.) consists of two sequential and dependent stages in training, but only contains one single model for recommendation, which is different from the sequential approach RoToR [6]. This is important because we may conduct offline training in two or more stages, but we have to provide recommendation on-the-fly in real deployment, which addresses the inconvenience challenge mentioned in Section III-B.

As another notice, we use compact and simple prediction rules defined on $U_{u\cdot}$, $V_{i\cdot}$, b_u and b_i in ROAF, i.e., $\hat{r}_{ui} = U_{u\cdot}V_{i\cdot} + b_i$ for ROAF(pai.) and $\hat{r}_{ui} = U_{u\cdot}V_{i\cdot} + b_i + b_u$ for ROAF(poi.), which makes the training procedures efficient in comparison with the integrative approach RoToR [6]. This addresses the inefficiency challenge in Section III-B.

V. Experiments

In this section, we conduct extensive empirical studies to verify our main hypothesis. Specifically, we believe that our ROAF is able to learn the model parameters in the browser stage and then refine them in the purchaser stage in an adaptive manner, which thus exploits the two different types of one-class feedback in a principled way.

Algorithm 2 The algorithm of ROAF(poi.).

// The browser stage
Input: $\mathcal{R}^{\mathcal{B}}$, $\mathcal{R}^{\mathcal{P}}$.
Output: Φ.
Initialization: Initialize model parameters Φ.
for $t = 1, \ldots, T$ **do**
  Randomly sample a set $\mathcal{R}^{\overline{\mathcal{B}\cup\mathcal{P}}}$ from $\mathcal{R}^{\mathcal{A}}$.
  for $t_2 = 1, \ldots, |\mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{B}} \cup \mathcal{R}^{\overline{\mathcal{B}\cup\mathcal{P}}}|$ **do**
    Randomly pick up a pair (u,i) from $\mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{B}} \cup \mathcal{R}^{\overline{\mathcal{B}\cup\mathcal{P}}}$.
    Calculate the gradients.
    Update the corresponding model parameters.
  end for
end for
// The purchaser stage
Input: $\mathcal{R}^{\mathcal{P}}$, Φ.
Output: $\tilde{\Phi}$.
Initialization: Initialize model parameters $\tilde{\Phi} = \Phi$.
for $t = 1, \ldots, T$ **do**
  Randomly sample a set $\mathcal{R}^{\overline{\mathcal{P}}}$ from $\mathcal{R}\backslash\mathcal{R}^{\mathcal{P}}$.
  for $t_2 = 1, \ldots, |\mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\overline{\mathcal{P}}}|$ **do**
    Randomly pick up a pair (u,i) from $\mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\overline{\mathcal{P}}}$.
    Calculate the gradients.
    Update the corresponding model parameters.
  end for
end for

A. Datasets and Evaluation Metrics

In our empirical study, we evaluate the performance of our ROAF and other methods on two large datasets, including MovieLens 10M [17] and Netflix[1]. MovieLens 10M and Netflix are two commonly used datasets in empirical studies of recommendation methods, where the former contains about 10 million numerical rating records from 71,567 users and 10,681 items, and the latter contains about 0.1 billion numerical rating records from 480,189 users and 17,770 items. We take the following steps for data preprocessing: (i) we first randomly take 60% (user, item, rating) triples and keep the (user, item) pairs with rating value 5 as purchases; (ii) we then divide them into three parts with equal size, one part for training, one part for validation, and the left part for test; and (iii) we finally take the remaining 40% triples and keep all the (user, item) pairs as browses. We repeat this procedure for three times in order to obtain three copies of data.

For performance evaluation, we use five commonly used ranking-oriented metrics in recommender systems and information retrieval [18], [19], including Precision@5, Recall@5, F1@5, NDCG@5 and 1-call@5.

B. Baselines and Parameter Configurations

For comparative study, we include two methods for one-class collaborative filtering (OCCF) and six methods for HOCCF:

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

TABLE II

RECOMMENDATION PERFORMANCE OF OUR ROAF AND OTHER METHODS ON MOVIELENS 10M (TOP) AND NETFLIX (BOTTOM). NOTICE THAT THE RESULTS OF BPR, MF(LOGISTICLOSS), ABPR, TJSL AND RBPR ARE COPIED FROM [6] FOR DIRECT COMPARISON, "-" DENOTES THE CASE THAT THE TRAINING PROCESS CAN NOT BE FINISHED WITHIN 168 HOURS, AND THE SIGNIFICANTLY BEST RESULTS ARE MARKED IN BOLD.

Recommendation method		Feedback	Precision@5	Recall@5	F1@5	NDCG@5	1-call@5
1-stage	BPR	$\mathcal{R}^{\mathcal{P}}$	0.0629$\pm$ 0.0002	0.0855$\pm$ 0.0006	0.0603$\pm$ 0.0003	0.0861$\pm$ 0.0004	0.2648$\pm$ 0.0017
	MF(LogisticLoss)	$\mathcal{R}^{\mathcal{P}}$	0.0688$\pm$ 0.0005	0.0963$\pm$ 0.0006	0.0672$\pm$ 0.0006	0.0963$\pm$ 0.0007	0.2881$\pm$ 0.0018
	ABPR	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0657$\pm$ 0.0009	0.0893$\pm$ 0.0017	0.0632$\pm$ 0.0009	0.0905$\pm$ 0.0014	0.2752$\pm$ 0.0039
	TJSL	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0669$\pm$ 0.0006	0.1006$\pm$ 0.0001	0.0679$\pm$ 0.0005	0.0958$\pm$ 0.0002	0.2864$\pm$ 0.0014
2-stage	RBPR	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0719$\pm$ 0.0013	0.0977$\pm$ 0.0017	0.0690$\pm$ 0.0014	0.0994$\pm$ 0.0020	0.2990$\pm$ 0.0050
	RMF(LogisticLoss)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0743$\pm$ 0.0004	0.1047$\pm$ 0.0011	0.0727$\pm$ 0.0004	0.1050$\pm$ 0.0005	0.3102$\pm$ 0.0019
	ROAF(pai.)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0814$\pm$ 0.0005	0.1119$\pm$ 0.0014	0.0786$\pm$ 0.0006	0.1129$\pm$ 0.0014	0.3347$\pm$ 0.0020
	ROAF(poi.)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	**0.0820**$\pm$ 0.0003	**0.1168**$\pm$ 0.0005	**0.0806**$\pm$ 0.0002	**0.1152**$\pm$ 0.0001	**0.3385**$\pm$ 0.0007
1-stage	BPR	$\mathcal{R}^{\mathcal{P}}$	0.0716$\pm$ 0.0007	0.0480$\pm$ 0.0005	0.0446$\pm$ 0.0005	0.0818$\pm$ 0.0011	0.2846$\pm$ 0.0022
	MF(LogisticLoss)	$\mathcal{R}^{\mathcal{P}}$	0.0732$\pm$ 0.0001	0.0535$\pm$ 0.0002	0.0483$\pm$ 0.0001	0.0848$\pm$ 0.0001	0.2938$\pm$ 0.0008
	ABPR	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	-	-	-	-	-
	TJSL	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	-	-	-	-	-
2-stage	RBPR	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0797$\pm$ 0.0002	0.0595$\pm$ 0.0004	0.0527$\pm$ 0.0003	0.0939$\pm$ 0.0003	0.3174$\pm$ 0.0011
	RMF(LogisticLoss)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0795$\pm$ 0.0005	0.0625$\pm$ 0.0003	0.0540$\pm$ 0.0003	0.0951$\pm$ 0.0005	0.3183$\pm$ 0.0013
	ROAF(pai.)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	0.0845$\pm$ 0.0002	0.0630$\pm$ 0.0001	0.0557$\pm$ 0.0001	0.0988$\pm$ 0.0002	0.3320$\pm$ 0.0007
	ROAF(poi.)	$\mathcal{R}^{\mathcal{P}}$, $\mathcal{R}^{\mathcal{B}}$	**0.0861**$\pm$ 0.0007	**0.0685**$\pm$ 0.0006	**0.0590**$\pm$ 0.0005	**0.1023**$\pm$ 0.0009	**0.3407**$\pm$ 0.0020

- BPR [8] is a representative pairwise preference learning method for modeling homogeneous one-class feedback such as purchases in OCCF;
- MF(LogisiticLoss) [15] is a matrix factorization method with Logistic loss function based on a pointwise preference assumption;
- ABPR [2] is an extended and adaptive version of BPR by learning a confidence weight for each browsing behavior in HOCCF;
- TJSL [3] is a recent transfer learning-based joint similarity model for HOCCF;
- RBPR [5] is a role-based extension of BPR for HOCCF by viewing the heterogeneous feedback of browses and purchases separately, which is a two-stage re-ranking based method;
- RMF(LogisiticLoss) is a role-based extension of M-F(LogisticLoss) with two sequential steps similar to that of RBPR; and
- ROAF(pai.) and ROAF(poi.) are the pairwise preference learning and pointwise preference learning variants of our ROAF, respectively.

Notice that we use the results of BPR, MF(LogisticLoss), ABPR, TJSL and RBPR from [6] for direct comparison. For parameter configurations in RMF(LogisiticLoss) and our ROAF, we follow [6] and fix the number of latent dimensions $d = 20$ and the learning rate $\gamma = 0.01$, and search the best value of the tradeoff parameters (i.e., $\alpha_u, \alpha_v, \beta_u, \beta_v$) from $\{0.001, 0.01, 0.1\}$ and the best iteration number T from $\{100, 500, 1000\}$ according to the performance of ND-CG@15 on the validation data. For RMF(LogisticLoss) and ROAF(poi.), we randomly sample three times the number of interacted (user, item) pairs from the corresponding un-interacted data [16]. Notice that in the purchaser stage of our ROAF, we adopt a mild approach when choosing the iteration number, i.e., $T \in \{10, 20, ..., 990, 1000\}$ for model adaptation and refinement.

C. Results

We report the recommendation performance of our ROAF as well as the baseline methods in Table II, from which we can have following observations:

- our ROAF performs significantly better than all the baseline methods in all cases, which shows the effectiveness of our proposed role-based adaptive factorization solution;
- for the comparison between the closely related one-stage methods and the two-stage methods, i.e., BPR vs. RBPR, and MF(LogisticLoss) vs. RMF(LogisticLoss), we can see that the role-based two-stage methods, i.e., RBPR and RMF(LogisticLoss), perform better than the corresponding one-stage methods, i.e., BPR and MF(LogisticLoss), respectively, which verifies the rationality of the role-based perspective in modeling the two types of one-class feedback in HOCCF;
- for the comparison between the related pairwise methods and pointwise methods, i.e., BPR vs. MF(LogisticLoss), RBPR vs. RMF(LogisticLoss), and ROAF(pai.) vs. ROAF(poi.), we can see that the pointwise versions, i.e., MF(LogisticLoss), RMF(LogisticLoss) and ROAF(poi.), perform better in most cases, which shows the effectiveness of the pointwise preference assumption and the Logistic loss function for HOCCF; and

TABLE III
RECOMMENDATION PERFORMANCE AND TRAINING TIME OF OUR ROAF AND ROTOR(POI.,INT.) ON MOVIELENS 10M AND NETFLIX. NOTICE THAT ROTOR(POI.,INT.) PERFORMS BETTER THAN ROTOR(PAI.,INT.), AND THE RESULTS OF ROTOR(POI.,INT.) ARE COPIED FROM [6].

Dataset	Method	Precision@5	Recall@5	F1@5	NDCG@5	1-call@5	Training time (s)
ML10M	RoToR(poi.,int.)	0.0811± 0.0004	0.1173± 0.0005	0.0805± 0.0004	0.1149± 0.0007	0.3361± 0.0013	8577
	ROAF(poi)	0.0820± 0.0003	0.1168± 0.0005	0.0806± 0.0002	0.1152± 0.0001	0.3385± 0.0007	4035
Netflix	RoToR(poi.,int.)	0.0837± 0.0006	0.0670± 0.0005	0.0575± 0.0004	0.0993± 0.0006	0.3333± 0.0016	240520
	ROAF(poi)	0.0861± 0.0007	0.0685± 0.0006	0.0590± 0.0005	0.1023± 0.0009	0.3407± 0.0020	12213

TABLE IV
RECOMMENDATION PERFORMANCE AND TEST TIME OF OUR ROAF AND ROTOR(POI.,SEQ.) ON MOVIELENS 10M AND NETFLIX. NOTICE THAT ROTOR(POI.,SEQ.) PERFORMS BETTER THAN ROTOR(PAI.,SEQ.), AND THE RESULTS OF ROTOR(POI.,SEQ.) ARE COPIED FROM [6].

Dataset	Method	Precision@5	Recall@5	F1@5	NDCG@5	1-call@5	Test time (s)
ML10M	RoToR(poi.,seq.)	0.0779± 0.0001	0.1066± 0.0006	0.0751± 0.0002	0.1110± 0.0004	0.3192± 0.0020	957
	ROAF(poi)	0.0820± 0.0003	0.1168± 0.0005	0.0806± 0.0002	0.1152± 0.0001	0.3385± 0.0007	60
Netflix	RoToR(poi.,seq.)	0.0915± 0.0003	0.0679± 0.0004	0.0605± 0.0004	0.1089± 0.0005	0.3511± 0.0014	17362
	ROAF(poi)	0.0861± 0.0007	0.0685± 0.0006	0.0590± 0.0005	0.1023± 0.0009	0.3407± 0.0020	869

- for the four one-stage methods and four two-stage methods, we can see that MF(LogisticLoss) and our ROAF(poi.) are the two best methods in the two categories, respectively, in terms of both effectiveness and efficiency.

Notice that our ROAF(pai.) and ROAF(poi.) are associated with one parametric model instead of two in the two-stage methods RBPR and RMF(LogisticLoss), which is another advantage of our adaptive transfer learning solution.

We further study the performance improvement from the browser stage to the purchaser stage in our ROAF. We report the results of both the pairwise and pointwise preference learning variants in Figure 2. Specifically, we denote the model after training in the browser stage as ROAF(poi.)-B and the method that refines the model parameters in the purchaser stage as ROAF(poi.)-B-P. From Figure 2, we can see that adaptively refining the model parameters using the purchase data improves the performance in all cases, which shows the effectiveness of our two-stage solution and the complementarity of the two types of one-class feedback.

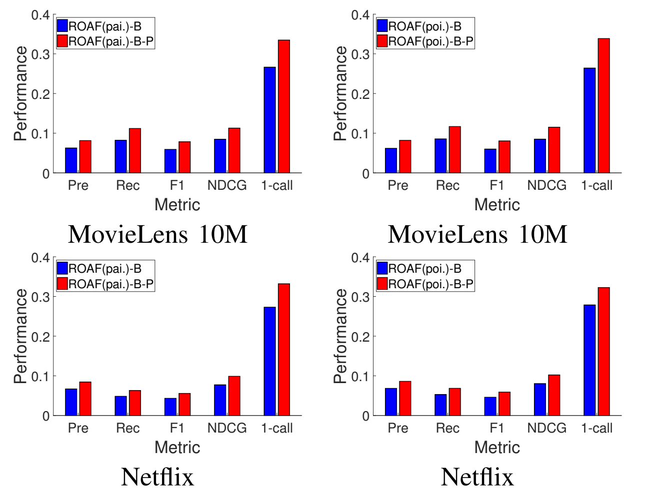

Fig. 2. Recommendation performance of our ROAF(pai.) and ROAF(poi.) after training in the browser stage and the purchaser stage, denoted as ROAF(pai.)-B, ROAF(pai.)-B-P, ROAF(poi.)-B, and ROAF(poi.)-B-P, on MovieLens 10M and Netflix.

D. The Efficiency Issue

RoToR is the state-of-the-art solution for HOCCF. However, the two variants in RoToR, i.e., RoToR(int.) and RoToR(seq.), suffer from some limitations as we mentioned before. In particular, the former will cause the inefficiency problem in training due to the complex expanded prediction rule, while the latter is inconvenient in deployment and maintenance because of the two parametric models and the candidate list of items. In this subsection, we conduct a quantitative study on this issue. Notice that the experiments are conducted on a machine with Intel i7-3370 3.40GHz CPU.

First of all, we compare the performance of accuracy and efficiency of RoToR(int.) and our ROAF on two datasets. Notice that we choose the pointwise variant, i.e., RoToR(poi.,int.), because it is better than the pairwise one, i.e., RoToR(pai.,int.), and report the results in Table III, where the results of RoToR(poi.,int.) are copied from [6]. We can see that our ROAF and RoToR(poi.,int.) are comparable on two datasets. Importantly, the training time of our ROAF is less than that of RoToR(poi.,int.), which is more significant on the larger dataset (i.e., Netflix).

As for the comparison between RoToR(seq.) and ROAF, we mainly focus on the test time since their training time is comparable for applying the sequential training style. The disadvantage of RoToR(seq.) is that it needs to maintain two different models, one for generating the candidate lists of items and the other for refining the lists of items, which increases the time in the prediction or test phase. On the contrary, our ROAF merely needs to maintain one single model though it is a two-stage solution in the training phase. Notice that we choose the pointwise variant, i.e., RoToR(poi.,seq.), because it is better than the pairwise one, i.e., RoToR(pai.,seq.), and report the results in Table IV, where the results of RoToR(poi.,seq.) are copied from [6]. The results show that our ROAF is better than RoToR(poi.,seq.) on ML10M but is worse on Netflix in terms of the recommendation accuracy. However, the test time of RoToR(poi.,seq.) is an order of magnitude longer than that of ROAF, and the gap becomes larger when the size of dataset increases. In other words, when the data becomes large, the prediction efficiency is improved by a large margin though sacrificing a bit of accuracy in our ROAF compared with RoToR, which is usually acceptable in real-world applications.

VI. Conclusions and Future Work

In this paper, we focus on modeling users' heterogeneous one-class feedback by adaptive knowledge transfer from a browser stage to a purchaser stage. Specifically, we design a novel solution termed as role-based adaptive factorization (ROAF) with a pairwise preference learning variant and a pointwise preference learning variant. In comparison with some recent integrative and sequential methods for HOCCF, i.e., RoToR(int.) and RoToR(seq.), our ROAF is a much more convenient and less complex solution in training and deployment due to the compact prediction rule and the single parametric model. As for other methods for modeling homogeneous or heterogeneous one-class feedback such as BPR, TJSL and others, our two variants of ROAF perform significantly better on two large datasets, which demonstrates the effectiveness of our adaptive transfer learning solution.

In the future, we are interested in further extending our adaptive transfer learning solution with deep learning [20], [21] and reinforcement learning paradigms [22], [23]. We are also interested in generalizing our two-stage solution to a multi-stage one by incorporating more one-class feedback such as "likes" and "collections", aiming to refine the model parameters more accurately.

Acknowledgment

We thank the support of National Natural Science Foundation of China Nos. 61872249, 61836005 and 61672358.

References

[1] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Proceedings of the 8th IEEE International Conference on Data Mining*, ser. ICDM'08, 2008, pp. 502–511.

[2] W. Pan, H. Zhong, C. Xu, and Z. Ming, "Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks," *Knowledge-Based Systems*, vol. 73, pp. 173–180, 2015.

[3] W. Pan, M. Liu, and Z. Ming, "Transfer learning for heterogeneous one-class collaborative filtering," *IEEE Intelligent Systems*, vol. 31, no. 4, pp. 43–49, 2016.

[4] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T.-S. Chua, D. Jin, and J. Yu, "Improving implicit recommender systems with view data," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, ser. IJCAI'18, 2018, pp. 3343–3349.

[5] X. Peng, Y. Chen, Y. Duan, W. Pan, and Z. Ming, "RBPR: Role-based Bayesian personalized ranking for heterogeneous one-class collaborative filtering," in *Late-breaking Results, Posters, Demos, Doctoral Consortium and Workshops Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalisation (UMAP 2016)*, 2016.

[6] W. Pan, Q. Yang, W. Cai, Y. Chen, Q. Zhang, X. Peng, and Z. Ming, "Transfer to rank for heterogeneous one-class collaborative filtering," *ACM Transactions on Information Systems*, vol. 37, no. 1, pp. 10:1–10:20, 2019.

[7] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, ser. NeurIPS'07, 2007, pp. 1257–1264.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, ser. UAI'09, 2009, pp. 452–461.

[9] H. Qiu, Y. Liu, G. Guo, Z. Sun, J. Zhang, and H. T. Nguyen, "BPRH: Bayesian personalized ranking for heterogeneous implicit feedback," *Information Sciences*, vol. 453, pp. 80 – 98, 2018.

[10] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. 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*, ser. KDD'18, 2018, pp. 1734–1743.

[11] M. Zhou, Z. Ding, J. Tang, and D. 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*, ser. WSDM'18, 2018, pp. 727–735.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2144–2155, 2019.

[14] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1381–1391, 2019.

[15] C. C. Johnson, "Logistic matrix factorization for implicit feedback data," in *NeurIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*, 2014.

[16] S. Kabbur, X. Ning, and G. 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*, ser. KDD'13, 2013, pp. 659–667.

[17] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 19:1–19:19, 2015.

[18] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[19] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells, "On the robustness and discriminative power of information retrieval metrics for top-N recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys'18, 2018, pp. 260–268.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 2017 World Wide Web Conference*, ser. WWW'17, 2017, pp. 173–182.

[21] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1754–1763.

[22] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "DRN: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW'18, 2018, pp. 167–176.

[23] J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, and J. Sun, "Hierarchical reinforcement learning for course recommendation in MOOCs," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, ser. AAAI'19, 2019.