Dual Similarity 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 focus on a recently studied problem in recommendation called heterogeneous one-class collaborative filtering (HOCCF), which consists of different types of feedback, e.g., target feedback and auxiliary feedback. The main challenge of HOCCF is how to make good use of the large-scale and ambiguous auxiliary feedback to assist the task of learning users' true preferences. Inspired by a recent method that jointly learns the similarity among items associated with the target feedback and auxiliary feedback in a transfer learning manner, we further extend this method by learning additional similarity among users for better modeling users' preferences. Specifically, we jointly learn the item-based similarity and user-based similarity by leveraging the target feedback and auxiliary feedback, and then aggregate the dual similarity into one single prediction rule for estimating users' preferences, which is thus termed as dual similarity learning model (DSLM). Extensive empirical studies on three public datasets show that our DSLM with dual similarity is more effective in terms of recommendation accuracy, demonstrating that our method strikes a good balance between the item-based similarity and user-based similarity.

Index Terms—Heterogeneous one-class collaborative filtering, dual similarity, matrix factorization

I. INTRODUCTION

Recommender systems play a significant role in providing personalized services for users to alleviate the problem of information overload. Previous methods in the past decades were proposed to model the users’ explicit feedback, e.g., ratings. But the research on recommendation has shifted to one-class collaborative filtering (OCCF) [1] in recent years. In OCCF problems, only one single type of feedback is available, e.g., purchases in e-commerce. Generally, we label the purchased records as “1” while the missing records as “0”. However, the performance of the methods for solving the OCCF problem is limited when the data are few, which is also called the sparsity problem. Therefore, heterogeneous one-class collaborative filtering (HOCCF) [2] is proposed to alleviate the above problem, in which more than one types of one-class feedback are available, e.g., purchases and examinations, with which we can improve the performance by exploiting the heterogeneous feedback.

HOCCF problems are common and significant in real-world applications. For instance, in e-commerce, users’ shopping patterns are usually clicks, collects, carts and purchases, among which purchases can be denoted as the target feedback and the remaining as the auxiliary feedback. Therefore, we can exploit the heterogeneous one-class feedback to extract the valuable information of users’ profiles for better recommendation. However, the main challenge is how to distinguish the valuable auxiliary feedback due to the ambiguity and noise. Specifically, a user may purchase an item just because it is a gift for his/her friends, or a user may collect an item and add it to cart but does not purchase it simply because of the high price. Therefore, it is essential to distinguish between the informative feedback and the noisy feedback.

Some recent works are proposed to address the problem of high uncertainty of the auxiliary feedback, e.g., TJSL [2]. Specifically, TJSL first identifies a set of items that users are likely to prefer with high confidence. And then, it learns a joint similarity among the items, i.e., a similarity between the target item and the previously purchased items, and a similarity between the target item and the examined items with high confidence. However, TJSL just considers the item-based similarity but ignores the user-based similarity, which might miss the valuable information.

In this paper, we propose a method called dual similarity learning model (DSLM), which further extends TJSL by learning an additional similarity among users for better modeling users’ preferences. Adopting the similar way to that of TJSL, we identify a set of potential users that are likely to purchase the target item, and then learn a joint similarity among the users like the way TJSL does. We aggregate the dual similarity into one single prediction rule for estimating users’ preferences. Experiments on three datasets show that the dual similarity designed for solving the HOCCF problem improves the recommendation performance, verifying the rationality of our method.

We summarize our main contributions of the paper as follows: (i) we extend a method called TJSL by designing a dual similarity, which is more effective for modeling heterogeneous one-class feedback; and (ii) we conduct extensive experiments on three public datasets, demonstrating that our method outperforms the state-of-the-art baseline methods.
II. PRELIMINARIES

A. Problem Definition

In this paper, we study the heterogeneous one-class collaborative filtering (HOCCF) problem. Suppose we have two types of feedback from \( n \) users and \( m \) items, including the target feedback of purchasing actions and the auxiliary feedback of examination actions. Specifically, for a user \( u \in \mathcal{U} \), we have a set of purchased items, i.e., \( \mathcal{P}_u \), and a set of examined items, i.e., \( \mathcal{E}_u \). Our goal is to exploit such two types of one-class feedback and recommend a ranked list of items for each user \( u \).

We list some commonly used notations and their explanations in Table I.

B. Factored Item Similarity Model

For homogeneous one-class collaborative filtering, factored item similarity model (FISM) [3] is a state-of-the-art method to capture users’ preferences by learning the similarity among items. Specifically, it factorizes the similarity matrix of items into low-dimensional feature vectors, with which we can estimate the similarity between two items via the inner product of their respective feature vectors, i.e., \( s_{ij} = \frac{1}{\sqrt{|\mathcal{P}_u| \cdot |\mathcal{P}_v|}} \mathbf{P}_u \cdot \mathbf{V}_v^T \).

Finally, we can estimate the preference of user \( u \) towards item \( i \) by aggregating the similarity between item \( i \) and all the items purchased by user \( u \) (i.e., \( \mathcal{P}_u \)),

\[
\sum_{v \in \mathcal{P}_u \setminus \{i\}} s_{vi} = \frac{1}{|\mathcal{P}_u \setminus \{i\}|} \sum_{v \in \mathcal{P}_u \setminus \{i\}} \mathbf{P}_v \cdot \mathbf{V}_i^T, \tag{1}
\]

where we can regard the term \( \frac{1}{|\mathcal{P}_u \setminus \{i\}|} \sum_{v \in \mathcal{P}_u \setminus \{i\}} \mathbf{P}_v \cdot \mathbf{V}_i^T \) as a certain virtual user profile w.r.t. the target feedback, denoting the distinct preference of user \( u \). Compared with the predefined similarity, e.g., Jaccard index or cosine similarity, factored item similarity is able to learn transitive relationships among items on sparse datasets to some degree with the assistance of low-dimensional feature vectors. However, the performance may be limited when the target feedback are few.

C. Transfer via Joint Similarity Learning

Inspired by FISM, transfer via joint similarity learning (TJSL) [2] is proposed to alleviate the sparsity problem by learning an additional similarity between a target item \( i \) and an examined item \( j \). To be specific, when the auxiliary feedback are available, TJSL introduces a new similarity term in a similar way to that of FISM, i.e., \( s_{ji} = \frac{1}{\sqrt{|\mathcal{E}_j| \cdot |\mathcal{E}_i|}} \mathbf{E}_j \cdot \mathbf{V}_i^T \), through which the knowledge from the auxiliary feedback (i.e., examinations) can be transferred. Then, the preference estimation of user \( u \) towards item \( i \) becomes as follows,

\[
\sum_{v \in \mathcal{P}_u \setminus \{i\}} s_{vi} + \sum_{j \in \mathcal{E}_u \setminus \{i\}} s_{ji}, \quad \mathcal{I}_u^c(t) \subseteq \mathcal{I}_u^c, \tag{2}
\]

where \( \sum_{j \in \mathcal{E}_u \setminus \{i\}} s_{ji} = \frac{1}{\sqrt{|\mathcal{E}_j| \cdot |\mathcal{E}_i|}} \sum_{j \in \mathcal{E}_u \setminus \{i\}} \mathbf{E}_j \cdot \mathbf{V}_i^T \), and the term \( \frac{1}{\sqrt{|\mathcal{P}_u| \cdot |\mathcal{P}_v|}} \sum_{v \in \mathcal{P}_u \setminus \{i\}} \mathbf{P}_v \cdot \mathbf{V}_i^T \) can be regarded as a virtual user profile w.r.t. the auxiliary feedback. Note that \( \mathcal{I}_u^c(t) \) is the set of likely-to-prefer items selected from \( \mathcal{I}_u^c \) due to the uncertainty of the auxiliary feedback. We can see that TJSL jointly learns the similarity between the target item and the purchased/examined items, mitigating the sparsity problem compared with FISM. But it is noticeable that both FISM and TJSL merely model the similarity among items, ignoring the similarity among users.

III. DUAL SIMILARITY LEARNING MODEL

In this section, we describe our method, i.e., dual similarity learning model (DSLM), which takes both item-based similarity and user-based similarity into account. Besides, we aggregate these similarity into one single prediction rule for joint filtering from the perspective of collective knowledge transfer. We will dive into the detail in the sequel.
A. Dual Similarity

In TJSL, two similarities among items are learned, i.e., the similarity $s_{ij}$ between a target item $i$ and a purchased item $i'$, and the similarity $s_{ji}$ between a target item $i$ and an examined item $j$. Symmetrically, we define the similarity among users as follows,

$$
\sum_{w' \in U_i \setminus \{u\}} s_{w'u} + \sum_{w \in U_i} s_{wu}, \quad U_i^{\mathcal{E}(t)} \subseteq \mathcal{U}_i^{E},
$$

(3)

where $\sum_{w' \in U_i \setminus \{u\}} s_{w'u} = \frac{1}{\sqrt{|U_i \setminus \{u\}|}} \sum_{w' \in U_i \setminus \{u\}} P_{w' \times U^T}$. $\sum_{w \in U_i} s_{wu} = \frac{1}{\sqrt{|U_i |}} \sum_{w \in U_i} E_w \times U^T$. Intuitively, we can also regard the term $\frac{1}{\sqrt{|U_i \setminus \{u\}|}} \sum_{w' \in U_i \setminus \{u\}} P_{w'}$ and $\frac{1}{\sqrt{|U_i |}} \sum_{w \in U_i} E_w$ as the virtual item profiles w.r.t. the target feedback and auxiliary feedback, respectively. $U_i^{\mathcal{E}(t)}$ is similar to that in TJSL, denoting the potential users that are likely to purchase item $i$.

With the user-based similarity, we aggregate all the similarity into one single prediction rule, which is inspired by hybrid collaborative filtering. The prediction rule is as follows,

$$
\hat{E}_i = \sum_{i' \in U_i^{\mathcal{E}(t)}} s_{i'i} + \sum_{j \in \mathcal{I}_i} s_{ji} + b_u + b_i + \sum_{w' \in U_i \setminus \{u\}} s_{wu}, \quad \mathcal{I}_i^{\mathcal{E}(t)} \subseteq \mathcal{I}_i^{E}, \quad \mathcal{I}_i^{E} \subseteq \mathcal{U}_i^{E}.
$$

(4)

We can see that the main difference between TJSL and our DSLM is that we introduce the user-based similarity in the prediction rule, composing the dual joint similarity that can better reflect users’ true preferences. The illustration of the dual similarity is shown in Fig.1.

B. Optimization Problem

Given the above prediction rule, we can learn the model parameters by optimizing the following objective function,

$$
\min_{\theta(\mathcal{E}), \mathcal{I}_i^{\mathcal{E}(t)} \subseteq \mathcal{I}_i^{E}, \mathcal{I}_i^{E} \subseteq \mathcal{U}_i^{E}} \sum_{(u,i) \in \mathcal{R}_P \cup \mathcal{R}_A} f_{ui}(t)
$$

(5)

where $f_{ui}(t) = \frac{1}{2}(r_{ui} - \hat{E}_i)^2 + \lambda_u \frac{1}{2} \|U_u\|^2 + \lambda_i \frac{1}{2} \|P_{i'}\|^2 + \frac{\lambda_f}{2} \sum_{w \in U_i} \|E_w\|^2 + \frac{\alpha}{2} \|V_i\|^2 + \frac{\alpha}{2} \sum_{j \in \mathcal{I}_i} \|P_{j'}\|^2 + \frac{\beta}{2} \sum_{j \in \mathcal{I}_i} \|E_j\|^2 + \frac{\gamma}{2} b_u^2 + \frac{\gamma}{2} b_i^2$, and the model parameters are $\Theta(\mathcal{E}) = \{U_u, P_{i'}, E_w, V_i, P_{j'}, E_j, b_u, b_i\}$. Note that $\mathcal{R}_A$ is a set of negative feedback used to complement the target feedback, where $r_{ui} = 1$ if $(u,i) \in \mathcal{R}_P$ and $r_{ui} = 0$ otherwise.

C. Learning Algorithm

To learn the parameters $\Theta(\mathcal{E})$, we use the stochastic gradient decent (SGD) algorithm and have the gradients of the model parameters for a randomly sampled pair $(u,i) \in \mathcal{R}_P \cup \mathcal{R}_A$,

$$
\nabla U_u = -e_u \frac{1}{\sqrt{|U_i^{\mathcal{E}(t)} \setminus \{u\}|}} \sum_{u' \in U_i^{\mathcal{E}(t)} \setminus \{u\}} P_{u'}, \quad -e_u \frac{1}{\sqrt{|U_i^{\mathcal{E}(t)} |}} \sum_{w \in U_i^{\mathcal{E}(t)}} E_w + \lambda_u U_u,
$$

(6)

$$
\nabla V_i = -e_u \frac{1}{\sqrt{|I_i^{\mathcal{E}(t)} \setminus \{u\}|}} \sum_{i' \in I_i^{\mathcal{E}(t)} \setminus \{i\}} \tilde{P}_{i'}, \quad -e_u \frac{1}{\sqrt{|I_i^{\mathcal{E}(t)} |}} \sum_{j \in I_i^{\mathcal{E}(t)}} \tilde{E}_j + \alpha_v V_i,
$$

(7)

$$
\nabla P_{i'} = -e_u \frac{1}{\sqrt{|U_i^{\mathcal{E}(t)} \setminus \{u\}|}} U_u + \lambda_{p_{i'}} P_{i'}, \quad u' \in U_i^{\mathcal{E}(t)},
$$

(8)

$$
\nabla E_w = -e_u \frac{1}{\sqrt{|U_i^{\mathcal{E}(t)} |}} U_u + \lambda_{e} E_w, \quad w \in U_i^{\mathcal{E}(t)},
$$

(9)

$$
\nabla \tilde{P}_{i'} = -e_u \frac{1}{\sqrt{|I_i^{\mathcal{E}(t)} \setminus \{i\}|}} V_i + \alpha_p \tilde{P}_{i'}, \quad i' \in I_i^{\mathcal{E}(t)} \setminus \{i\},
$$

(10)

$$
\nabla \tilde{E}_j = -e_u \frac{1}{\sqrt{|I_i^{\mathcal{E}(t)} |}} V_i + \alpha_e \tilde{E}_j, \quad j \in I_i^{\mathcal{E}(t)},
$$

(11)

$$
\nabla b_u = -e_u + \beta_u b_u, \quad \nabla b_i = -e_u + \beta_i b_i,
$$

(12)

where $e_u = r_{ui} - \hat{E}_i$ is the difference between the true preference and the predicted preference. With the above gradients, we can then update the model parameters via the update rule,

$$
\theta(\mathcal{E}) \leftarrow \theta(\mathcal{E}) - \gamma \nabla \theta(\mathcal{E}),
$$

(13)

where $\gamma$ is the learning rate.

The algorithm is shown in Algorithm 1, which contains $L$ iterations. In the $t$th epoch, we randomly sample a negative set and merge it with the target set. Then we randomly pick up a $(u,i)$ pair from the union set, calculate the gradients and update the parameters. We repeat the procedure for $T$ times in each epoch. Note that we identify $U_i^{\mathcal{E}(t)}$ and $I_i^{\mathcal{E}(t)}$ by the following way, i) for each user $u \in U_i^{\mathcal{E}}$, we estimate the preference for the target item $i$, i.e., $\hat{E}_i$, and take $\tau |U_i^{\mathcal{E}(t)}| (\tau \in (0,1])$ with the highest scores as the potential users that are likely to purchase the target item $i$; ii) for each $j \in I_i^{\mathcal{E}}$, similarly, we estimate the preference $\hat{E}_j$, and take $\tau |I_i^{\mathcal{E}(t)}|$ with the highest scores as the candidate items. Finally, we save the model and data of the last $L_0$ epochs. The estimated preference is the average value of $\hat{E}_i$, where $\ell$ ranges from $L - L_0 + 1$ to $L$.

IV. EXPERIMENTS

A. Datasets and Evaluation Metrics

For direct comparison, we use the three datasets from [2], including ML100K, ML1M and Alibaba2015. For performance evaluation, we adopt two commonly used ranking-oriented metrics, i.e., Precision@5 and NDCG@5.

B. Baselines and Parameter Settings

For comparative studies, we include all the three most competitive methods in [2], i.e., Bayesian personalized ranking
Algorithm 1 The algorithm of DSLM.

1: Input: $\mathcal{R}^p, \mathcal{T}^\ell, T, L, L_0, \rho, \gamma, \lambda_* \alpha_*, \beta_*$
2: Output: $U^\ell(\Theta^\ell), T^\ell(\Theta^\ell)$ and $\Theta^\ell, \ell = L-L_0+1, \ldots, L.$
3: Let $U^\ell(\Theta^\ell) = U^\ell, T^\ell(\Theta^\ell) = T^\ell, \tau = 1$
4: for $\ell = 1, \ldots, L$ do
5: Initialize the model $\Theta^\ell$
6: for $t = 1, \ldots, T$ do
7: Randomly sample $R^A \subset \mathcal{R} \cap \mathcal{R}^p$ with $|R^A| = \rho |\mathcal{R}^p|
8: for t_2 = 1, \ldots, |\mathcal{R}^p \cup R^A| do
9: Randomly pick up $(u, i) \in \mathcal{R}^p \cup R^A$
10: Calculate $r_{ui}^{(t)}$ via (4)
11: Calculate $\nabla \theta, \theta \in \Theta^\ell$ via (4)–(12)
12: Update $\theta, \theta \in \Theta^\ell$ via (13)
13: end for
14: end for
15: if $\ell > L - L_0$ then
16: Save the current model and data $(\Theta^\ell), U^\ell(\Theta^\ell), T^\ell(\Theta^\ell)$
17: end if
18: if $L > 1$ and $L > \ell$ then
19: $\tau \leftarrow \tau \times 0.9$
20: Select $T_{ui}^{\ell+1}$ with $|T_{ui}^{\ell+1}| = \tau |T_{ui}^{\ell}|$ for each $u$
21: Select $T_{ui}^{\ell+1}$ with $|T_{ui}^{\ell+1}| = \tau |T_{ui}^\ell|$ for each $i$
22: end if
23: end for

(BPR) [5], factored item similarity model (FISM) [3] and transfer via joint similarity learning (TJSL) [2]. Moreover, we also include a staged solution called role-based Bayesian personalized ranking (RBPR) [4].

For DSLM, we fix the number of latent dimension $d = 20$, the learning rate $\gamma = 0.01$ and sampling parameter $\rho = 3$, and search the tradeoff parameters from $\{0.001, 0.01, 0.1\}$ and the best iteration number $T$ from $\{100, 500, 1000\}$ via NDCG@5 performance, which is the same with that in [2]. The data and code used in the experiments are publicly available.

C. Results

We report the recommendation performance in Table II, from which we can have the following observations:

- In all cases, the methods for modeling heterogeneous one-class feedback, i.e., TJSL, RBPR and DSLM perform significantly better than the methods for modeling homogeneous one-class feedback, i.e., BPR and FISM, which shows the effectiveness of introducing the auxiliary feedback to assist the task of learning users’ preferences.
- RBPR performs better than TJSL in most cases, showing the usefulness of the designed two-stage learning paradigm in RBPR. It also reflects that TJSL might fail to fully learn users’ preferences to some extent.
- DSLM performs better than TJSL and RBPR in most cases, e.g., DSLM is the best on ML1M and Alibaba2015, and is comparable with TJSL on ML100K, which clearly shows the usefulness of the dual similarity in capturing the correlations between users and items.

V. Conclusions and Future Work

In this paper, we propose a novel solution, i.e., dual similarity learning model (DSLM), for a recent and important recommendation problem called heterogeneous one-class collaborative filtering (HOCF). In particular, we jointly learn the dual similarity among both users and items so as to exploit the complementarity well. Extensive empirical studies on three public datasets clearly show the effectiveness of our solution.

For future works, we are interested in further improving our DSLM by exploring pairwise ranking models [6] and deep learning models [7].

ACKNOWLEDGMENT

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

REFERENCES