Transfer Learning for Heterogeneous One-Class Collaborative Filtering

Weike Pan, Mengsi Liu, Zhong Ming
panweike@szu.edu.cn, liumengsi@email.szu.edu.cn, mingz@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University

Abstract

Various memory- and model-based collaborative filtering algorithms have been designed for multi-class feedbacks such as grade scores in the past two decades. Recently, one-class feedbacks such as positive feedbacks and implicit examinations have been recognized as a more pervasive and important source of information in many real recommendation systems. Previous works in this line mainly focus on homogenous one-class positive feedbacks such as “likes” in Facebook or “transactions” in Amazon, which may not capture a user’s true preferences well due to the sparsity of such data. In order to alleviate the sparsity problem, we study the positive feedbacks and implicit examinations simultaneously, which is coined as heterogeneous one-class collaborative filtering (HOCCF). Specifically, we design a novel transfer learning algorithm for HOCCF, i.e., transfer via joint similarity learning (TJSL), which jointly learns a similarity between a candidate item and a preferred item, and a similarity between a candidate item and an identified likely-to-prefer examined item. Joint similarity learning has the merit of being able to connect two seemingly not related items w.r.t. the sparse positive feedbacks only. Empirical studies on three real-world data sets show that our TJSL can recommend more accurately than the state-of-the-art methods.

Keywords: Transfer Learning, Collaborative Filtering, Heterogeneous One-Class Feedbacks

1 Introduction

Intelligent recommendation technology [2, 3, 4, 5, 9, 10, 12] has been embedded in various online applications that connect people, products and services such as E-commerce, video streaming and social media sites. Users’ feedbacks such as ratings, transactions and examinations have been recognized as critical information for learning users’ preferences. In the past two decades, most works have been focused on exploiting users’ multi-class feedbacks [4, 9, 12] such as grade scores for different levels of preferences. Recently, one-class feedbacks [2, 3, 5, 10] such as “likes” in Facebook and “transactions” in Amazon have attracted more attention from both researchers and practitioners because such positive feedbacks have become more popular than numerical ratings in many real systems.

However, one-class positive feedbacks is associated with a fundamental problem of data sparsity [8], for example, a user’s transaction records may not be sufficient to learn his/her true preferences. In a real system, besides a user’s positive feedbacks, there are usually lots of implicit examinations such as “clicks”, “browse” and “collections”. Such implicit examination data are related to users’ preference though are of high uncertainty as compared with positive feedbacks, because we may not infer that a user likes an item based on a single “click” action.

As a response to the sparsity problem of the positive feedbacks, we study different types of one-class feedbacks simultaneously in a single collaborative filtering algorithm, which is coined as heterogeneous one-class collaborative filtering (HOCCF) as shown in Figure 1. We first map the important HOCCF problem to the transfer learning [6] paradigm via taking positive feedbacks as target data and implicit examinations as auxiliary data, and then design a novel transfer learning algorithm to identify and integrate some likely-to-prefer examined items for each user. Specifically, we achieve knowledge transfer via joint similarity learning (TJSL), which not only learns a similarity between a candidate item and a preferred item as done in previous works [3] but also a similarity between a candidate item and a selected examined item.

Joint similarity learning has the merit of bridging two seemingly not related items w.r.t. positive feedbacks only, and thus alleviate the sparsity problem. We use a toy example in Figure 1 to illustrate its advantages. From Figure 1, we are not aware that B1 and B2 are related w.r.t. positive feedbacks (the left part of Figure 1) and may only recommend B3 to Grace based on item popularity. However, if we can leverage the implicit examinations, we can easily find the correlation between B1 and B2 (the middle part of Figure 1) and then recommend B2 to Grace in a similar way to that of item-based recommendation (the right part of Fig-
Figure 1: Illustration of sparse positive feedbacks (left), dense implicit examinations (middle), and item-based recommendation (right) in HOCCF.

We conduct empirical studies of such joint similarity learning on three real-world data sets and find that our solution can achieve much better performance than the state-of-the-art methods for one-class collaborative filtering.

We summarize our main contributions as follows, (i) we study and map the HOCCF problem to the transfer learning paradigm, (ii) we design a novel transfer learning algorithm via joint similarity learning, and build connections between a candidate item and a preferred/examined item in a principled way, and (iii) we conduct empirical studies and show that our transfer learning solution can recommend more accurately than the state-of-the-art methods.

Table 1: Some notations.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>user set, $u \in U$</td>
</tr>
<tr>
<td>$i$</td>
<td>item set, $i, i', j \in I$</td>
</tr>
<tr>
<td>$P$</td>
<td>positive feedbacks</td>
</tr>
<tr>
<td>$E$</td>
<td>implicit examinations</td>
</tr>
<tr>
<td>$R$</td>
<td>all (user, item) pairs</td>
</tr>
<tr>
<td>$A \subset R \setminus P$</td>
<td>sampled feedbacks</td>
</tr>
<tr>
<td>$d$</td>
<td>latent feature number</td>
</tr>
<tr>
<td>$V_i, P_{i'}, E_j \in \mathbb{R}^{1 \times d}$</td>
<td>latent feature vectors</td>
</tr>
<tr>
<td>$b_u \in \mathbb{R}$</td>
<td>user bias</td>
</tr>
<tr>
<td>$b_i \in \mathbb{R}$</td>
<td>item bias</td>
</tr>
<tr>
<td>$r_{ui} \in {1, 0}$</td>
<td>preference of $(u, i)$</td>
</tr>
<tr>
<td>$\hat{r}_{ui}$</td>
<td>prediction of $(u, i)$</td>
</tr>
<tr>
<td>$T, L, L_0$</td>
<td>iteration number</td>
</tr>
<tr>
<td>$\rho$</td>
<td>sampling parameter</td>
</tr>
<tr>
<td>$\alpha_u, \alpha_p, \alpha_e, \beta_u, \beta_e$</td>
<td>tradeoff parameters</td>
</tr>
</tbody>
</table>

2 Background

2.1 Problem Definition

In a typical recommendation system, we have a user set $U = \{u\}$ and an item set $I = \{i\}$. For those users and items, we usually have some different types of one-class feedbacks, including positive feedbacks such as likes and implicit examinations such as clicks. Specifically, for a user $u \in U$, we have a set of preferred items, i.e., $P_u$, and a set of examined items, i.e., $E_u$. Our goal is then to exploit such two types of one-class feedbacks and recommend a ranked list of items from $I \setminus P_u$ for each user $u$.

We list some notations in Table 1.

2.2 Factored Item Similarity Model

Factored item similarity model (FISM) [3] is a recent state-of-the-art recommendation method using homogeneous one-class feedbacks such as the positive feedbacks $P$ in HOCCF (i.e., the left part of Figure 1). The main idea of FISM is to factorize the similarity between a preferred item $i' \in P_u$ and a candidate item $i$ ($i \neq i'$) for user $u$ into two latent feature vectors with the same dimension, i.e., $P_{i'}, V_i \in \mathbb{R}^{1 \times d}$. The similarity can then be estimated via the inner product of those two feature vectors with certain normalization, i.e.,

$$s_{i'i} = \frac{1}{\sqrt{|P_u \setminus \{i\}|}} P_{i'} V_i^T,$$

where $|P_u \setminus \{i\}|$ is the number of items preferred by user $u$ excluding item $i$ because of the constraint $i \neq i'$. With the similarity between item $i$ and each $i' \in P_u \setminus \{i\}$, we can estimate the preference of user $u$ on item $i$ in a similar way to that of item-oriented memory-based collaborative filtering with all of its neighbors (i.e., $P_u \setminus \{i\}$) [3],

$$\sum_{i' \in P_u \setminus \{i\}} s_{i'i},$$

(1)
where \( \mathcal{P}_u \) is a set of items preferred by user \( u \). Note that the prediction rule of FISM also consists of a user bias \( b_u \) and an item bias \( b_i \), i.e., \( \hat{r}_{ui}^{\text{FISM}} = \sum_{i' \in \mathcal{P}_u \setminus \{i\}} s_{i'i} + b_u + b_i \).

FISM [3] boosts the recommendation performance over item-oriented memory-based collaborative filtering methods via learning the similarity instead of using some pre-defined similarity measurement such as Jaccard index or cosine similarity.

However, we may not learn the item-item similarity well when the positive feedbacks \( \mathcal{P} \) are few, because two items may not be well connected via the sparse positive feedbacks only as illustrated in the left part of Figure 1. This motivates us to leverage the relatively dense implicit examinations \( \mathcal{E} \) (the middle part of Figure 1) to improve the item-item similarity learning.

### 3 Transfer via Joint Similarity Learning

In this section, we describe our solution for HOCCF, i.e., transfer via joint similarity learning (TJSL), which aims to (i) identify and transfer some likely-to-prefer items for each user \( u \) from his/her implicitly examined items \( \mathcal{E}_u \), and (ii) jointly learn the similarities w.r.t. the positive feedbacks and the selected implicit examinations. Hence, TJSL has the potential to address the data sparsity problem of the target positive feedbacks via integrating the auxiliary implicit examinations from a transfer learning view.

#### 3.1 Model Formulation

In FISM [3], in order to estimate the preference of user \( u \) on item \( i \), we learn a similarity between item \( i \) and a preferred item \( i' \), i.e., \( s_{i'i} \) with \( i' \in \mathcal{P}_u \setminus \{i\} \), which is shown in Eq.(1). We go one step beyond and learn an additional similarity between item \( i \) and an examined item \( j \), i.e., \( s_{ji} \) with \( j \in \mathcal{E}_u \), which is expected to alleviate the sparsity problem of the positive feedbacks in HOCCF via transferring some preference knowledge from implicit examinations. With the positive feedbacks dependent similarity \( s_{i'i} \) and implicit examinations dependent similarity \( s_{ji} \), we further integrate them for preference estimation of user \( u \) on item \( i \),

\[
\sum_{i' \in \mathcal{P}_u \setminus \{i\}} s_{i'i} + \sum_{j \in \mathcal{E}_u} s_{ji},
\]

which shows that we jointly learn the similarities for knowledge transfer between implicit examinations and positive feedbacks. Hence, we call our approach transfer via joint similarity learning (TJSL).

However, due to the uncertainty of users’ preferences of the implicit examinations, we may not treat the examined items as we do for those preferred items. As a response, we propose to select some likely-to-prefer items \( \mathcal{E}_u^{(t)} \) from examined items \( \mathcal{E}_u \), and have

\[
\sum_{i' \in \mathcal{P}_u \setminus \{i\}} s_{i'i} + \sum_{j \in \mathcal{E}_u^{(t)}} s_{ji}, \quad \mathcal{E}_u^{(t)} \subseteq \mathcal{E}_u,
\]

With joint similarity, we reach a generic prediction rule,

\[
\hat{r}_{ui}^{(t)} = \sum_{i' \in \mathcal{P}_u \setminus \{i\}} s_{i'i} + \sum_{j \in \mathcal{E}_u^{(t)}} s_{ji} + b_u + b_i, \quad \mathcal{E}_u^{(t)} \subseteq \mathcal{E}_u,
\]

where \( b_u \) and \( b_i \) are commonly used to capture the user bias and item bias, respectively.

Due to the lack of negative feedbacks, we follow a common trick [3, 5] of randomly sampling some negative feedbacks \( \mathcal{A} \subseteq \mathcal{R} \setminus \mathcal{P} \) to complement the positive feedbacks. Furthermore, we assume that the preference of a positive feedback is \( r_{ui} = 1 \) with \( (u, i) \in \mathcal{P} \), and that of a negative feedback is \( r_{ui} = 0 \) with \( (u, i) \in \mathcal{A} \), which is usually called pointwise preference assumption [1, 3, 5].

Finally, with the above prediction rule in Eq.(4) and the sampled negative feedbacks \( \mathcal{A} \), we reach an optimization problem,

\[
\min_{\Theta(t), \mathcal{E}_u^{(t)} \subseteq \mathcal{E}_u} \sum_{(u, i) \in \mathcal{P} \cup \mathcal{A}} f_{ui}^{(t)},
\]

where \( f_{ui}^{(t)} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\epsilon}{2} ||P_i||_F^2 + \frac{\alpha_u}{2} \sum_{i' \in \mathcal{P}_u \setminus \{i\}} ||P_{i'}||_F^2 + \frac{\beta_u}{2} b_u^2 + \frac{\beta_i}{2} b_i^2 \), and the model parameters are \( \Theta(t) = \{b_u, b_i, V_i, P_{i'}, E_j|u \in \mathcal{U}, i \in \mathcal{I}, i' \in \mathcal{P}_u \setminus \{i\}, j \in \mathcal{E}_u^{(t)} \} \).

The differences between TJSL and FISM are two fold, including (i) the generic prediction rule and (ii) the identification of the likely-to-prefer items from implicit examinations.

#### 3.2 Learning the TJSL

In order to solve the optimization problem in Eq.(5), we have to learn the model parameters \( \Theta(t) \) and identify some likely-to-prefer examined items \( \mathcal{E}_u^{(t)} \), which will be answered separately in the sequel.

Given some selected examined items \( \mathcal{E}_u^{(t)} \), we have the gradient of each corresponding \( \theta \in \Theta(t) \) for \( (u, i) \in \mathcal{P} \cup \mathcal{A} \),

\[
\nabla \theta = \frac{\partial f_{ui}^{(t)}}{\partial \theta}, \quad (6)
\]

where \( \nabla \theta \) can be \( \nabla b_u = -e_{ui} + \beta_b b_u, \nabla b_i = -e_{ui} + \beta_b b_i, \nabla V_i = -e_{ui} \sqrt{||P_{i'}||_F^2} \sum_{i' \in \mathcal{P}_u \setminus \{i\}} P_{i'}, + \frac{1}{\sqrt{||E_j||^2 + \alpha V_{i'}} \sum_{j \in \mathcal{E}_u^{(t)}(j)} |E_j| + \alpha V_{i'}} \nabla E_{i'} = -e_{ui} \frac{1}{\sqrt{||P_{i'}||_F^2}} V_i + \alpha P_{i'}, i' \in \mathcal{P}_u \setminus \{i\}, \) and \( \nabla E_{i'} = -e_{ui} \frac{1}{\sqrt{||E_i||}} V_i + \).
1: **Input**: Positive feedbacks \( \mathcal{P} \), implicit examinations \( \mathcal{E} \), iteration numbers \( T, L, L_0 \), and parameters \( \rho, \alpha_u, \alpha_p, \beta_u, \beta_p \).

2: **Output**: Selected examinations \( \mathcal{E}(\ell) \) and learned models \( \Theta(\ell) \), \( \ell = L - L_0 + 1, \ldots, L \).

3: Let \( \mathcal{E}(1) = \mathcal{E} \), \( \tau = 1 \)

4: for \( \ell = 1, \ldots, L \) do
5: Initialize the model \( \Theta(\ell) \)
6: for \( t = 1, \ldots, T \) do
7: Randomly sample \( A \subset \mathcal{R}\setminus\mathcal{P} \) with \( |A| = \rho|\mathcal{P}| \)
8: for \( t_2 = 1, \ldots, |\mathcal{P} \cup A| \) do
9: Randomly pick up \((u, i) \in \mathcal{P} \cup A \)
10: Calculate \( \hat{r}_{ui}(\ell) \) via Eq.(4)
11: Calculate \( \nabla \theta, \theta \in \Theta(\ell) \) via Eq.(6)
12: Update \( \theta \), \( \theta \in \Theta(\ell) \) via Eq.(7)
end for
end for
13: end if
14: if \( \ell > L - L_0 \) then
15: Save the current model and data, i.e., \( \Theta(\ell), \mathcal{E}(\ell) \)
16: end if
17: if \( L > 1 \) and \( L > \ell \) then
18: \( \tau \leftarrow 0.9 \)
19: for \( u \in \mathcal{U} \) do
20: Select \( \mathcal{E}_{u}(\ell+1) \subseteq \mathcal{E}_{u} \) with \( |\mathcal{E}_{u}(\ell+1)| = \tau|\mathcal{E}_{u}| \)
21: end for
22: end for
23: end if
24: end for

Figure 2: The algorithm of transfer via joint similarity learning (TJSL).

\[ \alpha_j E_j, j \in \mathcal{E}_u. \]  
Note that \( e_{ui} = r_{ui} - \hat{r}_{ui} \) is the difference between the true preference and the estimated preference. Then, we have the update rule for each corresponding \( \theta \in \Theta(\ell) \),

\[ \theta \leftarrow \theta - \gamma \nabla \theta, \]  \hspace{1cm} (7)

where \( \gamma (\gamma > 0) \) is the learning rate.

Once we have learned the model parameters \( \Theta(\ell) \), we can identify some likely-to-prefer items from \( \mathcal{E}_u \) via the prediction rule in Eq.(4). Specifically, for each examined item \( j \in \mathcal{E}_u \) by user \( u \), we estimate its preference score \( \hat{r}_{ui}(\ell) \), and then take \( \tau|\mathcal{E}_u| \) examined items with highest scores. Note that \( \tau (0 < \tau \leq 1) \) is a parameter for item selection, which is initialized as 1 in the beginning and is then gradually decreased via \( \tau \leftarrow \tau \times 0.9 \) so as to ignore some unlikely-to-prefer items.

### 3.3 Algorithm

We describe the whole algorithm in Figure 2, which consists of \( L \) iterations of (i) learning the model parameters (lines 5-14), (ii) saving the current model and data (lines 15-17), and (iii) identifying some likely-to-prefer items from examined items (lines 18-23). Note that the inner iteration number \( T \) denotes the number of times that the algorithm samples a set of negative feedbacks [3].

From the algorithm in Figure 2, we can have the following observations, (i) when \( L = L_0 = 1 \), it reduces to the case using the whole set of implicit examinations without selection, and (ii) when \( \mathcal{E} = \emptyset \), it reduces to the FISM algorithm for OCCF with homogeneous positive feedbacks only.

Once we have selected the examined items and learned the corresponding model parameters, we can make a prediction for user \( u \) and item \( i \) via a linear combination of the last \( L_0 \) models,

\[ \hat{r}_{ui} = \sum_{\ell=L-L_0+1}^{L} \hat{r}_{ui}(\ell) / L_0, \] \hspace{1cm} (8)

where \( \hat{e}_{ui}(\ell) \) as shown in Eq.(4) is the prediction on \((u, i)\) using the preferred items \( \mathcal{P}_{ui} \), selected implicitly examined items \( \mathcal{E}_{ui}(\ell) \), and the corresponding \( \ell \)th model parameters \( \Theta(\ell) \).

The time complexity of FISM and TJSL is \( O(T(1 + \rho)|\mathcal{P}|d|\mathcal{P}_u|) \) and \( O(LT(1 + \rho)|\mathcal{P}|d \max(|\mathcal{P}_u|, |\mathcal{E}(\ell)|)) \), respectively, where \( |\mathcal{P}_u| \) and \( |\mathcal{E}_u| \) are average numbers of items w.r.t. a user’s positive feedbacks and selected examinations. We can see that the increase of time cost mainly comes from the outer iteration number \( L \) and the average number of selected examined items.

### 4 Experimental Results

Table 2: Description of the data sets used in the experiments, including numbers of users \( |\mathcal{U}| \), items \( |\mathcal{I}| \), positive feedbacks \( |\mathcal{P}| \), implicit examinations \( |\mathcal{E}| \), and test positive feedbacks \( |\mathcal{P}_{te}| \).

| Data set          | \( |\mathcal{U}| \) | \( |\mathcal{I}| \) | \( |\mathcal{P}| \) | \( |\mathcal{E}| \) | \( |\mathcal{P}_{te}| \) |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| MovieLens100K     | 943                  | 1682                | 9438                | 45285               | 2153                |
| MovieLens11M      | 6040                 | 3952                | 90848               | 400083              | 45075               |
| Alibaba2015       | 7475                 | 5257                | 9290                | 62659               | 2322                |
4.1 Data Sets and Evaluation Metric

**MovieLens100K** MovieLens100K\(^1\) is a popular data set used for various collaborative filtering research, which contains 100000 rating records from |\(\mathcal{U}\)| = 943 users and |\(\mathcal{I}\)| = 1682 items. There are two publicly available rating files, i.e., “ua.base” and “ua.test”, which contains 80000 and 20000 rating records, respectively. In order to simulate the HOCCF problem setting in Figure 1, we follow previous works [8] and preprocess the data as follows: first, we randomly take 50% (user, item, rating) triples from “ua.base” and only keep the (user, item) pairs with rating equal to 5 as positive feedbacks \(\mathcal{P}\); second, we take the remaining 50% (user, item, rating) triples from “ua.base” and keep all the (user, item) pairs as implicit examinations \(\mathcal{E}\); third, we take the (user, item) pairs with rating equal to 5 in “ua.test” as test positive feedbacks \(\mathcal{P}_{te}\).

**MovieLens1M** MovieLens1M is also a very popular data set used in empirical studies of recommendation algorithms, which contains 100000 rating records from 6040 users and 3952 items contained in the file “ratings.data”. Similarly, we preprocess the data as follows: first, we randomly divide the rating records in “ratings.data” into five parts with equal size; second, we take two parts, i.e., 40%, of (user, item, rating) triples and only keep the (user, item) pairs with rating equal to 5 as positive feedbacks; third, we take another two parts, i.e., 40%, of (user, item, rating) triples and keep all the (user, item) pairs as implicit examinations; finally, we take the remaining part, i.e., 20%, of (user, item, rating) triples and only keep the (user, item) pairs with rating equal to 5 as test positive feedbacks.

**Alibaba2015** Alibaba2015 is a real data with positive feedbacks (purchases) and implicit examinations (clicks)\(^3\). We preprocess the data via removing users and items with fewer than 3 associated positive feedbacks, and obtain 7475 users, 5257 items, 11612 positive feedbacks and 62659 implicit examinations. We then randomly take 80% positive feedbacks (purchases) and implicit examinations (clicks) as test positive feedbacks.

The statistics of the processed data sets\(^3\) are shown in Table 2.

**Evaluation Metrics** We adopt two commonly used evaluation metrics in collaborative recommendation and information retrieval, i.e., normalized discounted cumulative gain (NDCG) and precision, to evaluate the top-\(K\) recommended items:

\[
\text{NDCG@}\!K = \frac{1}{\log(K)} \sum_{u \in \mathcal{U}_{te}} \frac{1}{z_u} \text{DCG@}\!K_u,
\]

where \(z_u = \sum_{k=1}^{K} \delta(\text{\(k\)th recommended item} \in \mathcal{P}_u^{te})\), \(\delta(x) = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases}\), \(\mathcal{P}_u^{te}\) is the set of preferred items by user \(u\) in the test data, \(Z_u\) is the best \(\text{DCG@}\!K_u\) with \(\mathcal{P}_u^{te}\) in the beginning of the recommended list, and \(\mathcal{U}_{te}\) is the set of users in the test data.

\[
\text{Precision@}\!K = \frac{1}{|\mathcal{U}_{te}|} \sum_{u \in \mathcal{U}_{te}} \text{Precision@}\!K_u,
\]

where \(\text{Precision@}\!K_u = \frac{1}{K} \sum_{k=1}^{K} \delta(\text{\(k\)th recommended item} \in \mathcal{P}_u^{te})\).

4.2 Baselines and Parameter Settings

For comparative empirical studies, we include the following baselines:

- **Popularity-based ranking (PopRank)** is a simple but effective method for OCCF, which recommends the most popular items w.r.t. users’ positive feedbacks.
- **Item-oriented memory-based collaborative filtering (ICF)** is a classical recommendation method based on some predefined item-item similarity such as Jaccard index.
- **Bayesian personalized ranking (BPR)** [10] is a state-of-the-art recommendation algorithm for OCCF based on a pairwise preference assumption.
- **Factored item similarity model (FISM)** [3] aims to learn similarities between items based on homogeneous one-class feedbacks, which is reported to be very competitive.

We use \(\frac{|\{i|\langle u,i \rangle \in \mathcal{P}\}|-|\mathcal{P}|}{|\mathcal{U}||\mathcal{I}|}\) and \(\frac{|\{u|\langle u,i \rangle \in \mathcal{P}\}|-|\mathcal{P}|}{|\mathcal{U}||\mathcal{I}|}\) as initializations for user bias \(b_u\) and item bias \(b_i\), respectively. For each latent feature, we use a small random value \((r - 0.5) \times 0.01\) as its initialization, where \(r\) (0 \(\leq r < 1\)) is a random variable.

For ICF, we use Jaccard index as the similarity measurement and 20 neighbors in the prediction rule. For BPR, FISM and TJSL, we fix \(d = 20\), \(\gamma = 0.01\), and search the best values of the tradeoff parameters from \{0.001, 0.01, 0.1\} and the inner iteration number \(T\) of the stochastic algorithm from \{100, 500, 1000\} via the NDCG@5 performance. For TJSL, we fix the outer iteration number as \(L = 10\) and the number of models for final prediction as \(L_0 = 3\). For FISM and TJSL, we fix the sampling factor \(\rho = 3\) [3]. We then run the experiments with the searched parameter values for five times and report the average performance.

\(^1\)http://grouplens.org/datasets/movielens/
\(^2\)http://tianchi.aliyun.com/competition/introduction.htm
\(^3\)The data and source code used in the experiments can be downloaded at http://home.cse.ust.hk/~weikep/TL4HOCCF/
4.3 Results

We report the main results in Table 3, from which we can have the following observations,

• PopRank and ICF do not perform well on most cases as expected because of the recommendation strategy without personalization or the predefined similarity without learning.

• FISM is close to or better than the state-of-the-art pairwise preference learning method BPR, which shows the effectiveness of similarity learning and neighborhood-based prediction rule in FISM as compared with that of BPR; and

• TJSL performs significantly better than FISM, which shows the usefulness of the selected examined items and the effectiveness of integrating them in a joint similarity learning manner in the proposed transfer learning solution.

As mentioned before, we can see that the proposed transfer learning algorithm is a generic solution for HOCCF and OCCF. In particular, (i) when \( L = L_0 = 1 \), TJSL reduces to the case of leveraging the implicit examinations without selection, denoted as TJSL(1,1), and (ii) when \( E = \emptyset \), TJSL further reduces to FISM. We illustrate their relationships below,

\[
\text{TJSL} \xrightarrow{L=L_0=1} \text{TJSL}(1,1) \xrightarrow{E=\emptyset} \text{FISM}. \quad (9)
\]

In order to study the performance of FISM, TJSL(1,1) and TJSL in depth, we report the detailed performance in Figure 3. From Figure 3, we can see,

• TJSL(1,1) is better than FISM in most cases, in particular on MovieLens100K and MovieLens1M, which shows the usefulness of integrating the examined items in a joint similarity learning manner as shown in Eq.(2); and

• TJSL is better than TJSL(1,1) in most cases, which shows the effectiveness of the selection of likely-to-prefer examined items in the knowledge transfer procedure.

Interestingly, we can see that the performance ordering among TJSL, TJSL(1,1) and FISM is closely related to their relationships shown in Eq.(9), which supports the effectiveness of the proposed transfer learning algorithm from a theoretical perspective.

5 Related Work

5.1 Recommendation with One-Class Feedbacks

Various recommendation algorithms have been designed to exploit users’ feedbacks, including multi-class feedbacks [4, 9], binary-class feedbacks [7] and one-class feedbacks [3, 10]. One-class feedbacks have become popular in research communities [1, 3, 5, 8, 10] in recent years due to its pervasiveness in many real systems. Most algorithms exploiting one-class feedbacks focus on homogeneous positive feedbacks with pointwise [1, 5], pairwise [10] and list-wise [11] preference assumptions. A very recent work [8] generalizes a state-of-the-art pairwise preference learning method [10] and learns a confidence value of each implicit examination. However, it needs to estimate the preference for every (user, item) pair during the learning procedure, which may not be efficient for real deployment.

Our TJSL is a pointwise recommendation method for heterogeneous one-class feedbacks, which is thus different from previous works.

5.2 Recommendation via Similarity Learning

Similarities between users and/or items are essential for most recommendation algorithms, which is also closely related to the so-called “collective intelligence” and “social taste”. There are mainly two branches of those algorithms, one is based on similarity calculation and the other is based on similarity learning. For similarity calculation based methods, various similarity measurement have been designed and adopted, including Pearson correlation coefficient, cosine similarity, Jaccord index and some extensions and hybridizations [12]. For similarity learning based algorithms, different loss functions are optimized such as point-wise loss [3] and pairwise loss [10]. Similarity calculation based methods are usually less competitive in empirical studies, which is also supported by our experimental results.

The most closely related work to ours is FISM [3], which connects two items via learning similarities between a candidate item and preferred items with some latent features. Our TJSL jointly learns item-item similarities based on heterogeneous one-class feedbacks including both positive feedbacks and implicit examinations, which is novel and has not been exploited before.

In a summary, our TJSL is different from other works in terms of the above two dimensions, i.e., one-class feedbacks and similarity, which is summarized in Table 4. As compared with existing transfer learning works [6], our TJSL is also a novel algorithm that exploits joint similarity learning.
Table 3: Recommendation performance of TJSL and other methods on MovieLens100K, MovieLens1M and Alibaba2015 using Precision@5 and NDCG@5. The significantly best results are marked in bold ($p$ value < 0.03).

<table>
<thead>
<tr>
<th></th>
<th>PopRank</th>
<th>ICF</th>
<th>BPR</th>
<th>FISM</th>
<th>TJSL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ML100K</strong></td>
<td>Precision@5</td>
<td>0.0393±0.0000</td>
<td>0.0264±0.0000</td>
<td>0.0552±0.0006</td>
<td>0.0628±0.0015</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.0649±0.0000</td>
<td>0.0380±0.0000</td>
<td>0.0874±0.0020</td>
<td>0.1029±0.0017</td>
</tr>
<tr>
<td><strong>ML1M</strong></td>
<td>Precision@5</td>
<td>0.0692±0.0000</td>
<td>0.0812±0.0000</td>
<td>0.0928±0.0008</td>
<td>0.0971±0.0013</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.0839±0.0000</td>
<td>0.0949±0.0000</td>
<td>0.1121±0.0010</td>
<td>0.1189±0.0008</td>
</tr>
<tr>
<td><strong>Alibaba2015</strong></td>
<td>Precision@5</td>
<td>0.0018±0.0000</td>
<td>0.0056±0.0000</td>
<td>0.0050±0.0006</td>
<td>0.0046±0.0003</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.0056±0.0000</td>
<td>0.0151±0.0000</td>
<td>0.0138±0.0017</td>
<td>0.0126±0.0009</td>
</tr>
</tbody>
</table>

Figure 3: Top-K recommendation performance of FISM, TJSL(1,1) and TJSL.

Table 4: Summary of some related works.

<table>
<thead>
<tr>
<th>Similarity calculation</th>
<th>One-class feedbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>homogeneous</td>
</tr>
<tr>
<td>KUNN [12], etc.</td>
<td>FISM [3], etc.</td>
</tr>
</tbody>
</table>

6 Conclusions and Future Work

In this paper, we study a new and important recommendation problem called heterogeneous one-class collaborative filtering (HOCCF) containing positive feedbacks and implicit examinations as shown in Figure 1. Specifically, we first map the HOCCF problem to the transfer learning paradigm and then design a novel transfer learning algorithm, i.e., transfer via joint similarity learning (TJSL), in order to address the sparsity problem of positive feedbacks in a principled way. Our TJSL jointly learns similarities between a candidate item and preferred/examined items and is able to connect two seemingly not related items w.r.t. sparse positive feedbacks. Empirical studies on three real-world data sets show that our TJSL is much more accurate than the state-of-the-art methods that do not leverage the implicit examinations.

For future works, we are interested in generalizing joint similarity learning from the pointwise preference assumption to pairwise or even listwise ones, and improving joint similarity learning via taking implicit examinations as a special type of content information of both users and items.

Acknowledgement

We thank the editors and reviewers for their expert comments and constructive suggestions, and the support of Natural Science Foundation of Guangdong Province No. 2014A030310268, National Natural Science Foundation of China No. 61502307, 61170077, Grant of Shenzhen City No. KQCX20140519103756206 and Natural Science Foundation of SZU No. 201436. Zhong Ming is the corresponding author for this work.

References


[3] Santosh Kabbur, Xia Ning, and George Karypis. Fism: Factored item similarity models for top-n rec-


The Authors

Weike Pan is currently a lecturer (research oriented) with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include transfer learning, recommender systems, and statistical machine learning. He received the Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, China, in 2012. Contact him at panweike@szu.edu.cn.

Mengsi Liu is currently a master student in the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include intelligent recommendation technology, data mining and machine learning. She received the B.S. degree in Computer Science from the Huanggang Normal University, Huanggang, Hubei, China, in 2014. Contact her at liumengsi@email.szu.edu.cn.

Zhong Ming is currently a professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include software engineering and web intelligence. He received the Ph.D. degree in Computer Science and Technology from the Sun Yat-Sen University, Guangzhou, Guangdong, China, in 2003. He is the corresponding author. Contact him at mingz@szu.edu.cn.