Transfer Learning for Heterogeneous One-Class Collaborative Filtering

Weike Pan, Mengsi Liu, and Zhong Ming, College of Computer Science and Software Engineering, Shenzhen University

Intelligent recommendation technology is embedded in various online applications that connect people, products, and services, such as e-commerce, video streaming, and social media sites. User feedback, which includes ratings, transactions, and examinations, is recognized as critical information for learning user preferences. In the past two decades, most works have focused on exploiting multiclass feedback, such as grade scores for different levels of preferences. But more recently, one-class feedback, such as likes on Facebook and transactions on Amazon, has attracted attention from both researchers and practitioners alike because it’s more popular than numerical ratings in many real systems.

However, one-class positive feedback is associated with the fundamental problem of data sparsity—for example, a user’s transaction records might not be sufficient to learn his or her true preferences. Real systems include implicit examinations, such as clicks, browses, and collections. Such implicit examination data are related to user preference, although they’re highly uncertain compared to positive feedback because we can’t infer that a user likes an item based on a single click action.

In response to the sparsity problem of positive feedback, our work looks at different types of one-class feedback simultaneously in a single collaborative filtering algorithm, coined heterogeneous one-class collaborative filtering (HOCCF). We first mapped the HOCCF problem to the transfer learning paradigm by taking positive feedback as target data and implicit examinations as auxiliary data, and then designed a novel transfer learning algorithm to identify and integrate likely-to-prefer examined items for each user.

Specifically, we achieved 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, but also finds a similarity between a candidate item and a selected examined item. TJSL has the merit of bridging two seemingly unrelated items for positive feedback, thus alleviating the sparsity problem. Figure 1 uses a toy example to illustrate its advantages. In Figure 1, we aren’t aware that B1 and B2 are related with regard to positive feedback because we can’t infer that a user likes an item based on a single click action.

In Figure 1, we aren’t aware that B1 and B2 are related with regard to positive feedback because we can’t infer that a user likes an item based on a single click action.
Related Work in Recommendation with One-Class Feedback

Various recommendation algorithms have been designed to exploit user feedback, including multiclass feedback, \cite{1,2,3,4} binary-class feedback, \cite{1} and one-class feedback.\cite{4} One-class feedback has become popular in research communities in recent years due to its pervasiveness in many real systems. Most algorithms exploiting one-class feedback focus on homogeneous positive feedback with pointwise,\cite{5,6} pairwise,\cite{7} and list-wise\cite{8} preference assumptions. A very recent work\cite{9} generalizes a state-of-the-art pairwise preference learning method\cite{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 might not be efficient for real deployment.

Similarities between users and items are essential for most recommendation algorithms, which are also closely related to collective intelligence and social taste. There are primarily two branches of these algorithms, one based on similarity calculation and the other based on similarity learning. For similarity calculation-based methods, various similarity measurements have been designed and adopted, including the Pearson correlation coefficient, cosine similarity, the Jaccard index, and various extensions and hybridizations.\cite{11} For similarity learning-based algorithms, different loss functions are optimized such as pointwise\cite{12} and pairwise loss.\cite{13} 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 the factored item similarity model (FISM),\cite{14} which connects two items via learning similarities between a candidate item and preferred items with some latent features. Our transfer via joint similarity learning (TJSL) method jointly learns item-item similarities based on heterogeneous one-class feedback, including both positive feedback and implicit examinations, which is novel and hasn’t been exploited before.

References

where $P_u$ is a set of items preferred by user $u$. Note that FISM’s prediction rule also consists of a user bias $b_u$ and an item bias $b_i$, that is, $r_{ui}^{\text{FISM}} = \sum_{i' \in P_u \setminus \{i\}} s_{ii'} + b_u + b_i$.

FISM boosts recommendation performance over item-oriented memory-based collaborative filtering methods by learning the similarity instead of using some predefined similarity measurement such as the Jaccard index or cosine similarity. However, we might not learn item-item similarity when positive feedback $P$ are few because two items might not be well connected via sparse positive feedback alone, as illustrated in Figure 1. This motivates us to leverage relatively dense implicit examinations using some predefined similarity to improve item-item similarity via sparse positive feedback alone, as illustrated in Figure 1. This motivates us to leverage relatively dense implicit examinations $E$ (see the middle part of Figure 1) to improve item-item similarity learning.

**Transfer via Joint Similarity Learning**

Our solution for HOCCF—TJSL—aims to identify and transfer likely-to-prefer items for each user $u$ from his or her implicitly examined items $E_u$, as well as to jointly learn similarities from positive feedback and selected implicit examinations. Hence, TJSL has the potential to address the data sparsity problem in target positive feedback by integrating auxiliary implicit examinations from a transfer learning view.

**Model Formulation**

In FISM, to estimate user $u$’s preference for item $i$, we learn a similarity between item $i$ and a preferred item $i'$, that is, $s_{ii'}$ with $i' \in P_u \setminus \{i\}$, which is shown in Equation 1. We go one step beyond and learn an additional similarity between item $i$ and an examined item $j$, that is, $s_{ji}$ with $j \in E_u$, which is expected to alleviate the sparsity problem in HOCCF by transferring preference knowledge from implicit examinations. With positive feedback dependent similarity $s_{ii'}$ and implicit examinations dependent similarity $s_{ji}$, we further integrate them for preference estimation of user $u$ on item $i$:

$$r_{ui} = \sum_{i' \in P_u \setminus \{i\}} s_{ii'} + \sum_{j \in E_u} s_{ji}$$

which shows that we jointly learn the similarities for knowledge transfer between implicit examinations and positive feedback.

However, due to the uncertainty of user preferences in implicit examinations, we might not treat the examined items as we do for preferred items. As a response, we propose selecting some likely-to-prefer items $E^{(t)}_u$ from examined items $E_u$, resulting in

$$r_{ui}^{(t)} = \sum_{i' \in P_u \setminus \{i\}} s_{ii'} + \sum_{j \in E^{(t)}_u} s_{ji} + b_u + b_i$$

With joint similarity, we reach a generic prediction rule,

$$r_{ui}^{(t)} = \sum_{i' \in P_u \setminus \{i\}} s_{ii'} + \sum_{j \in E^{(t)}_u} s_{ji}$$

where $b_u$ and $b_i$ are commonly used to capture user bias and item bias, respectively. Due to the lack of negative feedback, we follow a common trick of randomly sampling negative feedback $A \subset \mathbb{R} \setminus P$ to complement the positive feedback. Furthermore, we assume that the preference of a positive feedback is $r_{ui} = 1$ with $(u, i) \in P$, and negative feedback is $r_{ui} = 0$ with $(u, i) \in A$, which is usually called pointwise preference assumption.

**Figure 1. Transfer via joint similarity learning (TJSL).** We aren’t aware of the items B1 and B2 being related with regard to positive feedback (left) and might only recommend B3 to Grace based on item popularity. However, if we leverage implicit examinations, we can easily find the correlation between B1 and B2 (middle) and then recommend B2 to Grace in a similar way to that of item-based recommendation (right).

| Table 1. Some notations and explanations for feedback and model. |
|-----------------|-----------------|
| $U$             | User set $u \in U$ |
| $I$             | Item set $i, i', j \in I$ |
| $P$             | Positive feedback |
| $P_u = \{(u, i) | (u, i) \in P\}$ | Preferred items by user $u$ |
| $E$             | Implicit examinations |
| $E_u = \{(u, i) | (u, i) \in E\}$ | Examined items by user $u$ |
| $R = \{(u, i)\}$ | All (user, item) pairs |
| $A \subset \mathbb{R} \setminus P$ | Sampled feedback |
| $T_u$ | Test positive feedback |
| $d \in \mathbb{R}$ | Latent feature number |
| $V_i, P_i, E_i \in \mathbb{R}^{1 \times d}$ | Latent feature vectors |
| $b_u \in \mathbb{R}$ | User bias |
| $b_i \in \mathbb{R}$ | Item bias |
| $r_{ui} \in [1, 0]$ | Preference of $(u, i)$ |
| $\tilde{r}_{ui}$ | Prediction of $(u, i)$ |
| $T, L, L_o$ | Iteration number |
| $\rho$ | Sampling parameter |
| $\alpha_u, \beta_p, \beta_a, \beta_b$ | Tradeoff parameters |
Finally, with the prediction rule in Equation 4 and the sampled negative feedback $\mathcal{A}$, we reach an optimization problem:

$$
\min_{\Theta^{(l)}, \mathcal{E}^{(l)}} \sum_{u,i \in \mathcal{P} \cup \mathcal{A}} f_w^{(l)}(u,i),
$$

where

$$
f_w^{(l)}(u,i) = \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha_u}{2} \| V_i \|_F^2 + \frac{\alpha_p}{2} \sum_{i' \in \mathcal{P}_u \setminus \{i\} \mid \ell \in \mathcal{E}_u \setminus \{i'\}} \| P_{i'} \|_F^2 + \frac{\alpha_e}{2} \sum_{i' \in \mathcal{E}_u \setminus \{i\} \mid \ell \in \mathcal{E}_u \setminus \{i'\}} \left( E_{i'} - \hat{E}_{i'} \right)^2,
$$

and

$$
\nabla E_i = -e_{ui} \frac{1}{\sqrt{\| E_i \|}} V_j, \quad j \in \mathcal{E}_u.
$$

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 $\Theta^{(l)}$:

$$
\Theta \leftarrow \Theta - \nabla \Theta,
$$

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

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

**Algorithm**

Figure 2 describes the whole algorithm, which consists of $L$ iterations of learning the model parameters (lines 5 to 14), saving the current model and data (lines 15 to 17), and identifying some likely-to-prefer items from examined items (lines 18 to 23). Note that the inner iteration number $T$ denotes the number of times that the algorithm samples a set of negative feedback.

From Figure 2, we can make the following observations: when $L = L_0 = 1$, it reduces to the case using the whole set of implicit examinations without selection, and when $\mathcal{E} = \emptyset$, it reduces to the FISM algorithm for HOCCF with homogeneous positive feedback only.

Once we select the examined items and learn 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:

$$
\hat{r}_{ui} = \sum_{\ell=L-L_0+1}^{L} \frac{2^{\ell}}{L_0},
$$

where $\frac{2^{\ell}}{L_0}$ as shown in Equation 4 is the prediction on $(u, i)$ using the predicted implicitly examined items $\mathcal{E}_u^{(\ell)}$, and the corresponding $\ell$th model parameters $\Theta^{(\ell)}$.

The time complexity of FISM and TJSL is $O(T(\ell + \rho) \| \mathcal{P} \| d \| \mathcal{E}_u \|)$ and $O(LT(\ell + \rho) \| \mathcal{P} \| d \max(\| \mathcal{P} \|, \| \mathcal{E}_u \|))$, respectively, where $\| \mathcal{P} \|$ and $\| \mathcal{E}_u \|$ are the average numbers of items with regard to a user’s positive feedback and selected examinations. We can see that the increase of time cost is due primarily to the outer iteration number $L$ and the average number of selected examined items.

**Experimental Results**

Next, we study the effectiveness of the proposed algorithm on some real data.

**Datasets and Evaluation Metrics**

In the empirical studies, we use three real datasets and two ranking-oriented evaluation metrics.

**MovieLens100K**. MovieLens100K (http://grouplens.org/datasets/movielens) is a popular dataset used in collaborative filtering research that contains 100,000 rating records from $| \mathcal{U} | = 943$ users and $| \mathcal{I} | = 1,682$ items. There are two publicly available rating files, “ua.base” and “ua.test,” which contain 80,000 and 20,000 rating records, respectively. To simulate the HOCCF problem setting in Figure 1, we follow previous works and preprocess the data as follows: first, we randomly take 50 percent (user, item, rating) triples from “ua.base” and only keep the (user, item) pairs with ratings equal to 5 as positive feedback $\mathcal{P}$. Then, we take the remaining 50 percent (user,
item, rating) triples from “ua.base” and keep all the (user, item) pairs as implicit examinations. Finally, we take the (user, item) pairs with ratings equal to 5 in “ua.test” as test positive feedback.

MovieLens1M. MovieLens1M is another popular dataset used in empirical studies of recommendation algorithms; it contains 1 million rating records from 6,040 users and 3,952 items in the file “ratings.data.” Similarly to that of MovieLens100K, we preprocess the data as follows: first, we randomly divide the records in “ratings.data” into five parts with equal size. Then, we take two parts—that is, 40 percent—of the (user, item) pairs with ratings equal to 5 as positive feedback. Next, we take another two parts—or 40 percent—of the triples and keep all the (user, item) pairs as implicit examinations. Finally, we take the remaining part—that is, 20 percent—of the (user, item, rating) triples and only keep the (user, item) pairs with ratings equal to 5 as test positive feedback.

Alibaba2015. Alibaba2015 (http://tianchi.aliyun.com/competition/introduction.htm) is real data with positive feedback (purchases) and implicit examinations (clicks). We preprocess the data by removing users and items with fewer than three associated positive feedback, and obtain 7,475 users, 5,257 items, 11,612 positive feedback, and 62,659 implicit examinations. We then randomly take 80 percent positive feedback (that is, 9,290) for training and the remaining 20 percent feedback (that is, 2,322) for test.

Table 2 shows statistics for the processed datasets (the data and source code used in the experiments can be downloaded at http://home.cse.ust.hk/~weikep/TL4HOCCF).

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

- NDCG@K = \( \frac{1}{|U^P|} \sum_{u \in U^P} \frac{1}{|Z_u|} \sum_{k=1}^K \delta \left( k \text{th recommended item } \in P^u_u \right) \times \log(k+1) \)

where DCG@K_u = \( \sum_{k=1}^K \frac{\delta \left( k \text{th recommended item } \in P^u_u \right)}{k} \)

and Z_u is the best DCG@K_u with P^u_u being the set of preferred items by user u in the test data.

Precision@K = \( \frac{1}{|U^P|} \sum_{u \in U^P} \frac{1}{|Z_u|} \sum_{k=1}^K \delta \left( k \text{th recommended item } \in P^u_u \right) \times \log(k+1) \)

Baselines and Parameter Settings

For comparative empirical studies, we include the following baselines:

- Popularity-based ranking (PopRank) is a simple but effective method for one-class feedback, which recommends the most popular items with regard to users’ positive feedback.

- Item-oriented memory-based collaborative filtering (ICF) is a classical recommendation method based on predefined item-item similarity such as the Jaccard index as a similarity measurement.

- Bayesian personalized ranking (BPR) is a state-of-the-art recommendation algorithm for one-class feedback based on a pairwise preference assumption.

- FISM aims to learn similarities between items based on homogeneous one-class feedback, which are reported to be very competitive.

We use ((i1,i2,i) ∈ P)/(|P|) − ((P)/|P|) as initializations for user bias b_u and item bias b_i, respectively. For each latent feature, we use a small random value \( r \) as its initialization, where \( r = 0.01 \) is a random variable.

For ICF, we use the Jaccard index as the similarity measurement and 20 neighbors for 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 \( \rho = 3 \). We then run the experiments with the searched parameter values five times and report the average performance.

Results

Table 3 shows our primary results, from which we can make the following observations:

- PopRank and ICF don’t perform well in most cases 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, and shows the effectiveness of similarity learning and neighborhood-based prediction rule in FISM as compared with that of BPR.

- 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 earlier, we can see that the proposed transfer learning algorithm is a generic solution for HOCCF. In particular, when \( L = L_0 = 1 \), TJSL reduces to the case of leveraging implicit examinations without selection, denoted as TJSL(1,1), and when \( E = \emptyset \), TJSL further reduces to FISM:

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

Figure 3 shows the detailed performance of FISM, TJSL(1, 1), and TJSL, where we can see that

- TJSL(1,1) is better than FISM in most cases, in particular in MovieLens100K and MovieLens1M, which shows the usefulness of integrating
the examined items in a joint similarity learning manner, 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, as shown in Equation 9, which supports the effectiveness of the proposed transfer learning algorithm from a theoretical perspective.

In future work, we’re interested in generalizing joint similarity learning from the pointwise preference assumption to pairwise or even list-wise assumption, ideally improving joint similarity learning by 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 we acknowledge the support of the National Natural Science Foundation of China nos. 61502307 and 61170077, the Natural Science Foundation of Guangdong Province nos. 2014A030310268 and 2016A030313038, the Grant of Shenzhen City no. KQXZ20140519103756206, and the Natural Science Foundation of SZU no. 201436. Zhong Ming is the corresponding author for this work.

References

The authors
Weike Pan is a lecturer in the College of Computer Science and Software Engineering at Shenzhen University, China. His research interests include transfer learning, recommender systems, and statistical machine learning. Pan received a PhD in computer science and engineering from the Hong Kong University of Science and Technology. Contact him at panweike@szu.edu.cn.

Mengsi Liu is a master’s student in the College of Computer Science and Software Engineering at Shenzhen University, China. Her research interests include intelligent recommendation technology, data mining, and machine learning. Liu received a BS in computer science from Huanggang Normal University, China. Contact her at liumengsi@email.szu.edu.cn.

Zhong Ming, corresponding author, is a professor in the College of Computer Science and Software Engineering at Shenzhen University, China. His research interests include software engineering and Web intelligence. Ming received a PhD in computer science and technology from Sun Yat-Sen University, China. Contact him at mingz@szu.edu.cn.