Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Neighborhood-enhanced transfer learning for one-class collaborative filtering

Wanling Cai[a,b], Jiongbin Zheng[a], Weike Pan[a,*], Jing Lin[a], Lin Li[a], Li Chen[b], Xiaogang Peng[a,*], Zhong Ming[a,*]

[a] College of Computer Science and Software Engineering, Shenzhen University, China
[b] Department of Computer Science, Hong Kong Baptist University, China

ARTICLE INFO

*Article history:*
Received 10 August 2018
Revised 6 January 2019
Accepted 10 March 2019
Available online xxx

Communicated by Shiliang Sun

*Keywords:*
Transfer learning
One-class collaborative filtering
Matrix factorization
Neighborhood-based recommendation

ABSTRACT

Recommender systems have become more prevalent in recent years for providing users with personalized services such as movie recommendation and news recommendation. In real-world scenarios, they are naturally thought of as one-class collaborative filtering (OCCF) problems because most behavioral data are users' interaction records, e.g., browses or clicks, which are referred to as one-class feedback or implicit feedback. In these problems, the sparsity of observed feedback and the ambiguity of unobserved feedback make it difficult to capture users' true preferences. In order to alleviate that, two well-known approaches have been proposed, including factorization-based methods aiming to learn the relationships between users and items via latent factors, and neighborhood-based methods focusing on similarities between users or items. However, these two types of approaches are rarely studied in one single framework or solution for OCCF. In this paper, we propose a novel transfer learning solution, i.e., transfer by neighborhood-enhanced factorization (TNF), which combines these two approaches in a unified framework. Specifically, we extract the local knowledge of the neighborhood information among users, and then transfer it to a global preference learning task in an enhanced factorization-based framework. Our TNF is expected to exploit the local knowledge in a global learning manner well. Extensive empirical studies on five real-world datasets show that our proposed solution can perform significantly more accurate than the state-of-the-art methods.



## 1. Introduction

Modern consumers are exposed to a myriad of product choices in online services, such as e-commerce websites and audio/video streaming services. A huge selection of products cannot meet various needs and tastes of each user, which emphasizes the prominence of recommender systems [1,2]. A widely used approach to the design of recommender systems is collaborative filtering (CF), which is based on analyzing copious information about users' behaviors and preferences. In many real-world scenarios, user behavioral data such as click or not-click are more common and easier to be obtained than the counter part of numerical rating data, which are usually called implicit [3] or one-class feedback [4]. Therefore, many researchers have turned to study the one-class collaborative filtering (OCCF) problem with one-class feedback aiming to rank items for each user rather than to predict users' ratings to items in the Netflix competition.

The one-class collaborative filtering problem has been studied for several years. Some previous works can be generally categorized into two classes: neighborhood-based methods [5] and factorization-based methods [6,7]. Neighborhood-based methods are used to compute the similarities between users or items, considering that similar users usually have similar behaviors and similar items often receive similar attention. This type of approach mainly contains two representative methods, including (i) user-oriented collaborative filtering [8], which is developed to predict the preference of users to items on the basis of the records of the like-minded users, and (ii) item-oriented collaborative filtering [9], which explains that users are likely to prefer items similar to those they have observed before. Both these two methods are local in their nature because they only consider a small set of close neighbors.

On the contrary, factorization-based methods employ a global viewpoint to characterize users and items. In general, they model

* Corresponding authors.
*E-mail addresses:* cswlcai@comp.hkbu.edu.hk (W. Cai), jiongbin92@gmail.com (J. Zheng), panweike@szu.edu.cn (W. Pan), linjing4@email.szu.edu.cn (J. Lin), lilin20171@email.szu.edu.cn (L. Li), lichen@comp.hkbu.edu.hk (L. Chen), pengxg@szu.edu.cn (X. Peng), mingz@szu.edu.cn (Z. Ming).

https://doi.org/10.1016/j.neucom.2019.03.016
0925-2312/

users' preferences based on low-dimensional latent factors. Some of the successful design of latent factor models are premised on matrix factorization, such as Bayesian personalized ranking (BPR) [10] and factored item similarity model (FISM) [11]. Specifically, the focus of BPR is to take pairs of items as basic units and learn a latent representation of both users and items via maximizing the likelihood of pairwise preferences over the observed items and unobserved items. FISM is an item-based method that learns the (item, item) similarity matrix as the product of two low-dimensional latent factor matrices, which implicitly helps to learn transitive relations between items and performs better than most other recommendation methods. So far, methods based on matrix factorization have received a significant amount of attention due to their effectiveness and efficiency [12,13].

In this paper, we propose a novel transfer learning [14] solution, i.e., transfer by neighborhood-enhanced factorization (TNF), in which we extract the local knowledge of neighborhood and then transfer it into a global preference learning task for each user via matrix factorization techniques. In this way, we study these two methods in one single framework, expecting to inherit the merits of the localized neighborhood-based method and the globalized factorization-based method. More importantly, this factored representation of users and items allows TNF to capture and model transitive relations within a group of close neighbors on datasets of low density. We conduct empirical studies on five real-world datasets in order to verify our proposed solution. Experimental results show that our solution can indeed improve the recommendation accuracy.

In Section 2, we introduce some related recommendation approaches, including neighborhood-based methods, factorization-based methods and deep learning based methods. In Section 3, we formulate the studied one-class collaborative filtering problem and describe our proposed transfer learning solution in detail. In Section 4, we present our experimental results and associated analysis. In Section 5, we conclude this work with some future directions.

## 2. Related work

In this section, we review some classical approaches, including (i) neighborhood-based recommendation, (ii) factorization-based recommendation, and (iii) deep learning based recommendation.

### 2.1. Neighborhood-based recommendation

In industry, neighborhood-based recommendation methods have been effectively deployed, the mechanism of which is to relate users to new items by following chains of (user, item) adjacency. Neighborhood-based methods contain two specific types of approaches, i.e., user-oriented methods [8] and item-oriented methods [9]. User-oriented methods mainly focus on the like-minded users with similar behaviors, i.e., purchasing or browsing similar items, while item-oriented methods estimate users' preferences on the basis of the similarities between the target item and items that they have observed before. For user-oriented and item-oriented methods, neighborhood construction is always an essential step, which employs some similarity measurement between two users or two items. For implicit feedback, we have similarities such as cosine similarity, Jaccard index and their extensions [9]. For instance, in user-oriented approaches, the weighted positive feedback to an item by the most close neighbors of a target user $u$ is used to provide a prediction $\hat{r}_{ui}$ as follows,

$$\hat{r}_{ui} = \sum_{w \in \mathcal{U}_i \cap \mathcal{N}_u} s_{uw}, \tag{1}$$

where $s_{uw}$ is the similarity between user $u$ and user $w$, and $\mathcal{U}_i$ and $\mathcal{N}_u$ denote the set of users interacted with item $i$ and the set of nearest neighbors of user $u$, respectively. They are not only easy and intuitive to implement, but also can generate relatively precise results. In addition to the user-oriented and item-oriented collaborative filtering methods, some researchers have also unified these two methods with the notion of similarity fusion [15].

### 2.2. Factorization-based recommendation

In recent years, factorization-based methods have been widely investigated by many researchers and have also been well recognized as the state-of-the-art methods in most recommendation problems. A typical factorization-based method associates users and items with vectors of latent features, i.e., matrix factorization (MF) [3,16], which estimates (user, item) interactions in the low-dimensional latent space. In this way, the sparsity problem of recommender systems can be alleviated to some extent. Among factorization-based methods, the resulting dot product, i,e., $U_{u\cdot}V_{i\cdot}^T$, captures the global interest of the user $u$ in connection with the item $i$. Studies also show that most recommender systems perform better if user biases and item biases are taken into account [7]. Hence, the overall prediction rule can be written as follows,

$$\hat{r}_{ui} = b_u + b_i + U_{u\cdot}V_{i\cdot}^T. \tag{2}$$

For one-class collaborative filtering, factored item similarity model (FISM) is proposed to improve the predefined similarity in neighborhood-based methods by learning the similarities among items. It learns the (item, item) similarity matrix as a product of two low-dimensional latent factor matrices, which allows FISM to capture and model relations between items on sparse data to a certain degree. Formally, the prediction rule is as follows,

$$\hat{r}_{ui} = b_u + b_i + \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot}V_{i\cdot}^T, \tag{3}$$

where $W_{i'\cdot}$ and $V_{i\cdot}$ denote the embedding vectors for item $i'$ and $i$, respectively. Given the well-defined predictive model of Eqs. (2) and (3), one can learn model parameters by optimizing loss functions for recommendation, such as point-wise classification loss [4], pairwise regression loss [10], and listwise ranking loss [17]. In addition, some auxiliary information from other domains can also be incorporated into factorization-based methods [18,19]. In our work, we adopt a simple but novel way to incorporate the knowledge learned from the neighborhood information into the factorization-based methods, which is rarely studied in one single framework.

### 2.3. Deep learning based recommendation

Deep learning techniques have become increasingly important these days due to their excellent performance. Deep neural networks show their ability to learn underlying features from data, being well known for extracting and representing high-level abstractions from low-level original data, such as audio data [20] and image data [21]. For recommendation, DNN has been mainly employed for modelling auxiliary information, such as textual description [22] and acoustic features of music data [23]. While in recent years, some existing works [24] have explored deep learning models for recommendation based on implicit feedback and explicit ratings. The collaborative denoising auto-encoder (CDAE) [25] is one of the eye-catching work for improving CF with implicit feedback. It learns distributed representation of the users and items using a denoising auto-encoder structure and additionally plugs a user node to the input of auto-encoders for reconstructing the users' preferences. Distinct from CDAE, NeuMF is proposed by Xiangnan He et al. [26] and it adopts a two pathway architecture

**Table 1**
Some notations and their explanations.

| | |
|---|---|
| $n$ | number of users |
| $m$ | number of items |
| $u \in \mathcal{U}$ | user ID |
| $i, i' \in \mathcal{I}$ | item ID |
| $\mathcal{R} = \{(u, i)\}$ | universe of all possible (user, item) pairs |
| $\mathcal{P} = \{(u, i)\}$ | the whole set of observed (user, item) pairs |
| $\mathcal{A}, \lvert\mathcal{A}\rvert = \rho\lvert\mathcal{P}\rvert$ | a sampled set of negative feedback from $\mathcal{R} \backslash \mathcal{P}$ |
| $\mathcal{I}_u$ | item set observed by user $u$ |
| $d$ | number of latent dimensions |
| $b_u \in \mathbb{R}$ | user bias |
| $b_i \in \mathbb{R}$ | item bias |
| $V_{i\cdot} \in \mathbb{R}^{1\times d}$ | item-specific latent feature vector |
| $X_{u'\cdot} \in \mathbb{R}^{1\times d}$ | user-specific latent feature vector |
| $\mathcal{N}_u$ | a set of nearest neighbors of user $u$ |
| $\hat{r}_{ui}$ | predicted preference of user $u$ to item $i$ |
| $\alpha_v, \alpha_x, \beta_u, \beta_v$ | trade-off parameters on the regularization terms |
| $\gamma$ | learning rate |
| $T$ | iteration number in the algorithm |

to model (user, item) interactions with a multi-layer feed-forward neural network. The authors also devise three user-based CF models. Moreover, deep matrix factorization model [13] is proposed to learn a common low-dimensional space for the representation of users and items with a deep architecture.

## 3. Transfer by neighborhood-enhanced factorization

### 3.1. Problem definition

In one-class collaborative filtering with only positive feedback such as browses or clicks, we have $n = |\mathcal{U}|$ users, $m = |\mathcal{I}|$ items, and the corresponding one-class feedback in the form of (user, item) pairs denoted as $\mathcal{P} = (u, i)$. Our goal is to learn users' preferences from such behavioral data and recommend each user $u$ a personalized ranked list of items from $\mathcal{I} \backslash \mathcal{P}_u$, which represents the not yet observed items for each target user $u \in \mathcal{U}$.

We put some commonly used notations and their explanations in Table 1.

### 3.2. Neighborhood construction

In order to extract the local knowledge from the records of users' behaviors, we first select the neighbors for each user via a commonly used similarity measurement. More specifically, we calculate the cosine similarity between user $u$ and user $w$ as follows,

$$s_{uw} = \frac{|\mathcal{I}_u \cap \mathcal{I}_w|}{\sqrt{|\mathcal{I}_u|}\sqrt{|\mathcal{I}_w|}}, \tag{4}$$

where $|\mathcal{I}_u|$, $|\mathcal{I}_w|$, $|\mathcal{I}_u \cap \mathcal{I}_w|$ denote the number of items observed by user $u$, user $w$, and both user $u$ and user $w$, respectively. In our solution, we adopt cosine similarity mainly for two reasons. Firstly, cosine similarity is a normalized version of Jaccard index. Secondly, the performance of cosine similarity is comparable or sometimes better than that of Jaccard index in our experience. Once the similarity of each user pair has been calculated, we can obtain a set of the most similar users of each user $u$ to construct a neighborhood $\mathcal{N}_u$. This neighborhood denotes the local knowledge of the users.

### 3.3. Neighborhood-enhanced factorization

In real-world scenarios, the records of feedback such as browses or clicks provided by users to items make an extremely small proportion of the (user, item) matrix, which results in the so-called sparsity problem of the training data. In order to ease this problem, methods based on matrix factorization are employed to capture global relationships between users and items by projecting the original feedback data to a low-dimensional space, while neighborhood-based methods concentrate on a small set of close neighbors to calculate the similarities between users or items. By the comparison between these two types of methods, we find that we could take a further step to integrate their merits to better address the existing problem.

With the intention of making full use of the two types of the aforementioned recommendation methods, we propose a novel transfer learning solution, which transfers the extracted knowledge of close neighbors into a general global optimization framework. In our TNF shown in Fig. 1, we assume that the knowledge of neighborhood extracted from the local association can be incorporated into a global factorization framework so as to better capture the latent representation. This process is just as human learning, in which people with intense concentration would digest knowledge locally but effectively while others with a big picture in mind are experts in building correlations between different domains or tasks. We make an assumption that the learners who are able to exploit a key combination of the local and global cues may make a greater achievement. Hence, for the studied OCCF problem, we bridge the localized neighborhood-based method with the globalized factorization-based method in an enhanced factorization-based framework. Specifically, a recent work [27] inspires us to aggregate the like-minded users' preferences. Finally, we have the estimated preference of user $u$ to item $i$ as follows,

$$\hat{r}_{ui} = b_u + b_i + \frac{1}{\sqrt{|\mathcal{N}_u|}} \sum_{u' \in \mathcal{N}_u} X_{u'\cdot} V_{i\cdot}^T. \tag{5}$$

The variables in the prediction rule in Eq. (5) are the user bias $b_u \in \mathbb{R}$, the item bias $b_i \in \mathbb{R}$, the user-specific latent feature vector $X_{u'\cdot} \in \mathbb{R}^{1\times d}$, and the item-specific latent vector $V_{i\cdot} \in \mathbb{R}^{1\times d}$. Notice that $\frac{1}{\sqrt{|\mathcal{N}_u|}} \sum_{u' \in \mathcal{N}_u} X_{u'\cdot}$ aggregates the preferences of close neighbors to represent user $u$'s interest. We replace the user-specific latent factor of user $u$ with this aggregated one in the prediction rule, for which we call it a virtual user-specific feature vector. In this way, the local knowledge of neighborhood can be transferred into the factorization-based method. For this reason, we call it *transfer by neighborhood-enhanced factorization* (TNF). Such a modeling approach shows improvement of recommendation accuracy in our empirical studies that will be discussed in Section 4. The local knowledge of neighborhood are clearly one salient source of information, and the effectiveness of TNF here demonstrates the importance of the second phase of our transfer learning solution. This implies that the local knowledge of users' neighborhood are transferred to the global learning task in a way that ensures the factorization-based method can learn users' preferences with deep concentration as well as global association.

Notice that one key difference between our solution and FISM [11] is that we learn a latent representation of users and items by transferring the neighborhood knowledge of user $u$ (i.e., $\mathcal{N}_u$), while FISM focuses on learning the factored item similarity by incorporating the knowledge of items that have been observed by user $u$ (i.e., $\mathcal{I}_u$).

### 3.4. Loss function

In our TNF, we adopt pointwise preference learning as our preference learning paradigm, which is more flexible than the pairwise one. The focal point of the latter is modeling the preference difference of a user $u$ to two items $i$ and $j$ [10], i.e., $\hat{r}_{ui} - \hat{r}_{uj}$, instead of modeling the preferences $\hat{r}_{ui}$ and $\hat{r}_{uj}$ separately in pointwise preference learning. With the pointwise preference assumption, we encode the probability of a user $u$ choosing to interact with an item

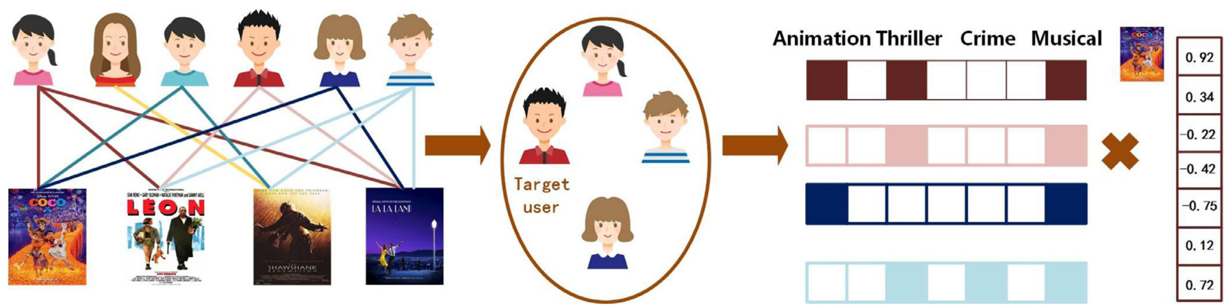



**Fig. 1.** Illustration of our transfer learning solution, i.e., transfer by neighborhood-enhanced factorization, for one-class collaborative filtering (OCCF). Our TNF consists of two phases, one of which is to extract the local knowledge by constructing a neighborhood for each user, and the other captures the latent representation of the users and items via a factorization-based global preference learning task.

$i$ as:

$$p(r_{ui}=1|\Theta)=\sigma(\hat{r}_{ui}), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function:

$$\sigma(z)=\frac{1}{1+e^{-z}}. \tag{7}$$

We can have the log-likelihood for an observed pair $(u,i)\in\mathcal{P}$ and an unobserved pair $(u,i)\in\mathcal{A}$ as follows,

$$\log\prod_{(u,i)\in\mathcal{P}\cup\mathcal{A}}p(r_{ui}=1|\Theta)^{r_{ui}}(1-p(r_{ui}=1|\Theta))^{1-r_{ui}}, \tag{8}$$

which will encourage to find the parameters that maximize the preference of user $u$ to an observed item $i$ and minimize the preference of user $u$ to an unobserved item $i$. With Eq. (6), we can rewrite the log-likelihood in Eq. (8) as follows:

$$-(r_{ui}\log(1+\exp(-\hat{r}_{ui}))+(1-r_{ui})\log(1+\exp(\hat{r}_{ui})). \tag{9}$$

In our solution, instead of using $r_{ui}=1$ and $r_{ui}=0$ to denote positive and negative preference for an observed $(u,i)\in\mathcal{P}$ pair and an unobserved $(u,i)\in\mathcal{A}$ pair, respectively, we equivalently convert them to $r_{ui}=1$ and $r_{ui}=-1$ for simplicity. Then, we have the following formula:

$$-\log(1+\exp(-r_{ui}\hat{r}_{ui})). \tag{10}$$

Combining all possible observed $(u, i)$ pairs for each user $u\in\mathcal{U}$, we reach the overall log-likelihood:

$$\sum_{(u,i)\in\mathcal{P}\cup\mathcal{A}}-\log(1+\exp(-r_{ui}\hat{r}_{ui})). \tag{11}$$

### 3.5. Objective function

Maximizing the overall log-likelihood in Eq. (11) is equivalent to solve the following optimization problem:

$$\min_{\Theta}\sum_{(u,i)\in\mathcal{P}\cup\mathcal{A}}f_{ui}+\mathcal{R}(\Theta), \tag{12}$$

where $f_{ui}=\log(1+\exp(-r_{ui}\hat{r}_{ui}))$ is the loss function we have mentioned before, and $\Theta=\{X_{u'\cdot},V_{i\cdot},b_u,b_i\}, i,i'=1,\ldots,m, u,u'=1,\ldots,n$ denotes the model parameters to be learned. In addition, we introduce the regularization term $\mathcal{R}(\Theta)=\frac{\alpha_x}{2}\sum_{u'\in\mathcal{N}_u}||X_{u'\cdot}||_F^2+\frac{\alpha_v}{2}||V_{i\cdot}||_F^2+\frac{\beta_u}{2}b_u^2+\frac{\beta_v}{2}b_i^2$ so that it can contribute to avoiding overfitting, where $\alpha_x$, $\alpha_v$, $\beta_u$and $\beta_v$ are trade-off hyper parameters.

### 3.6. Gradients and update rules

In order to solve the optimization problem in Eq. (12), we adopt the commonly used stochastic gradient descent (SGD) algorithm. Specifically, we calculate the gradients, i.e., $\nabla X_{u'\cdot}$, $\nabla V_{i\cdot}$, $\nabla b_u$ and $\nabla b_i$ for each $(u,i)\in\mathcal{P}\cup\mathcal{A}$ as follows,

$$\nabla X_{u'\cdot}=\frac{\partial f_{ui}}{\partial X_{u'\cdot}}=-e_{ui}\frac{1}{\sqrt{|\mathcal{N}_u|}}V_{i\cdot}+\alpha_x X_{u'\cdot}, u'\in\mathcal{N}_u, \tag{13}$$

$$\nabla V_{i\cdot}=\frac{\partial f_{ui}}{\partial V_{i\cdot}}=-e_{ui}\frac{1}{\sqrt{|\mathcal{N}_u|}}\sum_{u'\in\mathcal{N}_u}X_{u'\cdot}+\alpha_v V_{i\cdot}, \tag{14}$$

$$\nabla b_u=\frac{\partial f_{ui}}{\partial b_u}=-e_{ui}+\beta_u b_u, \tag{15}$$

$$\nabla b_i=\frac{\partial f_{ui}}{\partial b_i}=-e_{ui}+\beta_v b_i, \tag{16}$$

where $e_{ui}=\frac{r_{ui}}{1+\exp(r_{ui}\hat{r}_{ui})}$, and $\bar{U}_{u\cdot}=\frac{1}{\sqrt{|\mathcal{N}_u|}}\sum_{u'\in\mathcal{N}_u}X_{u'\cdot}$ is a certain virtual user-specific latent feature vector of user $u$ aggregated from the set of user neighborhood $\mathcal{N}_u$. For each $(u,i)\in\mathcal{P}\cup\mathcal{A}$, we have the update rules,

$$X_{u'\cdot}=X_{u'\cdot}-\gamma\nabla X_{u'\cdot}, u'\in\mathcal{N}_u, \tag{17}$$

$$V_{i\cdot}=V_{i\cdot}-\gamma\nabla V_{i\cdot}, \tag{18}$$

$$b_u=b_u-\gamma\nabla b_u, \tag{19}$$

$$b_i=b_i-\gamma\nabla b_i, \tag{20}$$

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

### 3.7. Algorithm

We depict the learning algorithm in Algorithm 1 and we can see that our TNF contains two phases. In the first phase, we construct a neighborhood for each user via a neighborhood-based method. In the second phase, we have two loops. In the outer loop, we randomly sample a set of not yet observed items for each user in order to construct an expanded set of users' behaviors with both positive feedback and negative feedback, i.e., $\mathcal{P}\cup\mathcal{A}$. In the inner loop, we update the model parameters based on each randomly drawn $(u, i)$ pair from $\mathcal{P}\cup\mathcal{A}$, which is more efficient than the user-wise sampling strategy in [28].

## 4. Experimental results

### 4.1. Datasets and evaluation metrics

In our empirical studies, we evaluate the performance of our proposed TNF on five real-world datasets with different densities,

**Algorithm 1** The algorithm of transfer by neighborhood-enhanced factorization (TNF).

```
Input: Observations P
Output: Recommended items for each user
Initialize model parameters Θ
Construct a neighborhood N_u for each user u
for t_1 = 1, ..., T do
    Randomly pick up a set A with |A| = ρ|P|
    for t_2 = 1, 2, ..., |P ∪ A| do
        Randomly draw a (u, i) pair from P ∪ A
        Calculate Ū_u· = 1/sqrt(|N_u|) Σ_{u'∈N_u} X_{u'·}
        Calculate r̂_ui = b_u + b_i + Ū_u· V_i·^T
        Calculate e_ui = r_ui / (1+exp(r_ui r̂_ui))
        Update b_u, b_i, V_i· and X_{u'·} for u' ∈ N_u
    end for
end for
```

**Table 2**
Statistics of the datasets used in the experiments, including the number of users ($n$), the number of items ($m$), the number of (user, item) pairs in the training data ($|\mathcal{P}|$), the number of (user, item) pairs in the test data ($|\mathcal{P}^{te}|$), and the density of each data i.e., $(|\mathcal{P}|+|\mathcal{P}^{te}|)/n/m$.

| Dataset | $n$ | $m$ | $\lvert\mathcal{P}\rvert$ | $\lvert\mathcal{P}^{te}\rvert$ | $(\lvert\mathcal{P}\rvert+\lvert\mathcal{P}^{te}\rvert)/n/m$ |
|---|---|---|---|---|---|
| ML100K | 943 | 1682 | 27,688 | 27,687 | 3.49% |
| ML1M | 6040 | 3952 | 287,641 | 287,640 | 2.41% |
| UserTag | 3000 | 2000 | 123,218 | 123,218 | 4.11% |
| Netflix5K5K | 5000 | 5000 | 77,936 | 77,936 | 0.62% |
| XING5K5K | 5000 | 5000 | 39,197 | 39,197 | 0.31% |

namely MovieLens[1] 100K and 1M, UserTag [4], a subset of Netflix[2] and a subset of XING[3], which have been used in [28,29]. MovieLens 100K (ML100K) and MovieLens 1M (ML1M) are the subsets of data from the MovieLens research project, UserTag is about the users' tagging behaviors, and Netflix5K5K and XING5K5K are the subsets of the data extracted from the Netflix Prize dataset and the XING job recommendation contest dataset, respectively. For ML100K, ML1M and Netflix5K5K, we only retain the ratings larger than 3 as the observed positive feedback to simulate the one-class feedback. For UserTag, the tagging behaviors are taken as one-class feedback. For XING5K5K, we take click, bookmark and reply as positive feedback. For our five datasets, we randomly sample half of the observed (user, item) pairs as training data, and the rest as test data. Through randomly taking one (user, item) pair for each user on average from the training data, we construct a validation set. Then we repeat the above procedure for three times. As such, we have three copies of training data, validation data and test data. The experimental results are averaged over the performance on those three copies of test data. The characteristics of all the datasets are summarized in Table 2. Notice that the datasets and code are publicly available.[4]

In prediction, we will recommend a ranked list of items for each user $u$ as generated by the recommendation algorithms. The ranked list is sorted by the predicted preference scores of user $u$ to all the not yet observed items, i.e., $\mathcal{I}\backslash\mathcal{P}_u$, from which we select the first $k$ items and denote them as $\mathcal{I}_u^{re}$. According to the recommendation results, we can then calculate the following ranking-oriented evaluation metrics.

- **Precision** is used to show the ratio of true predictions in the recommendation list:

$$\text{Prec}@k = \frac{1}{|\mathcal{U}^{warm}|}\sum_{u\in\mathcal{U}^{warm}}\frac{1}{k}\sum_{l=1}^{k}\delta(i(l)\in\mathcal{I}_u^{te}). \tag{21}$$

where $\mathcal{U}^{warm}=\mathcal{U}^{tr}\cap\mathcal{U}^{te}$ denotes the set of warm-start users who appear both in the training data and the test data, $\mathcal{I}_u^{te}$ denotes the item set preferred by user $u$ in the test data, $i(l)$ represents the $l$th item in the recommendation list, and $\delta(x)$ is an indicator function with the value 1 when x is true and the value 0 when x is false.

- **Recall** is used to describe the ratio of truly predicted items in the test set,

$$\text{Rec}@k = \frac{1}{|\mathcal{U}^{warm}|}\sum_{u\in\mathcal{U}^{warm}}\frac{1}{|\mathcal{I}^{te}|}\sum_{l=1}^{k}\delta(i(l)\in\mathcal{I}_u^{te}). \tag{22}$$

- **F1** combines precision and recall:

$$\text{F1}@k = 2\times\frac{\text{Prec}@k\times\text{Rec}@k}{\text{Prec}@k+\text{Rec}@k}. \tag{23}$$

- **NDCG** is mainly used to emphasize the ranked positions of the items in the list:

$$\text{NDCG}@k = \frac{1}{|\mathcal{U}^{warm}|}\sum_{u\in\mathcal{U}^{warm}}\frac{1}{Z_u}\text{DCG}_u@k \tag{24}$$

where $\text{DCG}_u@k=\sum_{l=1}^{k}\frac{2^{\delta(i(l)\in\mathcal{I}_u^{te})}-1}{\log(l+1)}$, and $Z_u$ is the best $\text{DCG}_u@k$ score.

- **1-call** is used to show whether there is at least one item truly predicted in the recommendation list:

$$\text{1-call}@k = \frac{1}{|\mathcal{U}^{warm}|}\sum_{u\in\mathcal{U}^{warm}}\delta\Big(\big(\sum_{l=1}^{k}\delta(i(l)\in\mathcal{I}_u^{te})\big)>0\Big). \tag{25}$$

For each evaluation metric, we first calculate the performance for each user from the test data, and then obtain the averaged performance over all the users in the test data.

### 4.2. Baselines and parameter settings

In our experiments, we use the following three closely related baseline algorithms for the studied problem:

- UCF (user-oriented collaborative filtering) [8] is a typical neighborhood-based method for OCCF. We use the cosine similarity as the similarity measurement for every two users when implementing this method.
- MF (matrix factorization) [7] is the most basic factorization-based method which learns the latent representation of users and items. We use square loss as our loss function when implementing this method.
- BPR (Bayesian personalized ranking) [10] is one of the most accurate recommendation algorithms for OCCF, which captures users' preferences according to the assumption that a user prefers an observed item to an unobserved one.
- FISM (factored item similarity model) [11] is an item-based method and it learns the similarity among items in latent factors, which aims to improve the predefined similarity in neighborhood-based methods for OCCF.
- NeuMF (neural matrix factorization) [26] is built on deep neural network model and adopts a two pathway architecture, including generalized matrix factorization (GMF) and multi-layer perceptron for recommendation tasks.

Our TNF (transfer by neighborhood-enhanced factorization) extracts the local knowledge of neighborhood via cosine similarity and then transfers it to a globalized factorization-based framework.

[1] https://grouplens.org/datasets/movielens/.
[2] https://www.netflix.com/.
[3] https://www.netflix.com/.
[4] http://csse.szu.edu.cn/staff/panwk/publications/TNF/.

Please cite this article as: W. Cai, J. Zheng and W. Pan et al., Neighborhood-enhanced transfer learning for one-class collaborative filtering, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.03.016

**Table 3**
Recommendation performance of user-oriented collaborative filtering (UCF) [8], matrix factorization (MF) [7], Bayesian personalized ranking (BPR) [10], factored item similarity models (FISM) [11], neural matrix factorization (NeuMF) [26] and our transfer by neighborhood-enhanced factorization (TNF) on five real-world datasets. The significantly best results are marked in bold.

| Dataset | Method | Prec@5 | Rec@5 | F1@5 | NDCG@5 | 1-call@5 |
|---|---|---|---|---|---|---|
| ML100K | UCF | 0.3448 ± 0.0020 | 0.0867 ± 0.0012 | 0.1197 ± 0.0011 | 0.3647 ± 0.0049 | 0.7940 ± 0.0197 |
| | MF | 0.3669 ± 0.0086 | 0.0983 ± 0.0045 | 0.1348 ± 0.0053 | 0.3842 ± 0.0078 | 0.8303 ± 0.0107 |
| | BPR | 0.3504 ± 0.0065 | 0.0915 ± 0.0043 | 0.1274 ± 0.0041 | 0.3670 ± 0.0069 | 0.8082 ± 0.0161 |
| | FISM | 0.4011 ± 0.0032 | 0.1009 ± 0.0011 | 0.1401 ± 0.0011 | 0.4161 ± 0.0037 | 0.8370 ± 0.0059 |
| | NeuMF | 0.3648 ± 0.0085 | 0.0936 ± 0.0048 | 0.1293 ± 0.0054 | 0.3789 ± 0.0094 | 0.8057 ± 0.0176 |
| | TNF | **0.4118** ± 0.0080 | **0.1052** ± 0.0031 | **0.1452** ± 0.0042 | **0.4316** ± 0.0084 | **0.8538** ± 0.0134 |
| ML1M | UCF | 0.3705 ± 0.0026 | 0.0615 ± 0.0011 | 0.0942 ± 0.0012 | 0.3855 ± 0.0024 | 0.8090 ± 0.0004 |
| | MF | 0.4174 ± 0.0005 | 0.0704 ± 0.0007 | 0.1080 ± 0.0005 | 0.4306 ± 0.0010 | 0.8437 ± 0.0045 |
| | BPR | 0.4180 ± 0.0039 | 0.0665 ± 0.0008 | 0.1030 ± 0.0011 | 0.4300 ± 0.0040 | 0.8202 ± 0.0049 |
| | FISM | 0.4241 ± 0.0013 | 0.0727 ± 0.0005 | 0.1114 ± 0.0005 | 0.4388 ± 0.0018 | 0.8478 ± 0.0046 |
| | NeuMF | 0.3995 ± 0.0105 | 0.0658 ± 0.0019 | 0.1011 ± 0.0026 | 0.4143 ± 0.0100 | 0.8176 ± 0.0105 |
| | TNF | **0.4602** ± 0.0044 | **0.0781** ± 0.0011 | **0.1193** ± 0.0013 | **0.4781** ± 0.0040 | **0.8662** ± 0.0019 |
| UserTag | UCF | 0.2524 ± 0.0028 | 0.0400 ± 0.0005 | 0.0624 ± 0.0013 | 0.2619 ± 0.0028 | 0.5757 ± 0.0093 |
| | MF | 0.2957 ± 0.0022 | 0.0456 ± 0.0012 | 0.0722 ± 0.0015 | 0.3032 ± 0.0024 | 0.6146 ± 0.0077 |
| | BPR | 0.2883 ± 0.0034 | 0.0439 ± 0.0012 | 0.0695 ± 0.0016 | 0.2959 ± 0.0039 | 0.5978 ± 0.0009 |
| | FISM | 0.2797 ± 0.0089 | 0.0413 ± 0.0022 | 0.0658 ± 0.0031 | 0.2871 ± 0.0064 | 0.5686 ± 0.0055 |
| | NeuMF | 0.2943 ± 0.0076 | 0.0462 ± 0.0008 | 0.0731 ± 0.0013 | 0.3021 ± 0.0086 | 0.6049 ± 0.0100 |
| | TNF | **0.3195** ± 0.0018 | **0.0513** ± 0.0013 | **0.0802** ± 0.0013 | **0.3320** ± 0.0030 | **0.6367** ± 0.0014 |
| Netflix5K5K | UCF | 0.1939 ± 0.0016 | 0.0657 ± 0.0013 | 0.0780 ± 0.0008 | 0.2112 ± 0.0029 | 0.5221 ± 0.0026 |
| | MF | 0.2239 ± 0.0029 | 0.0935 ± 0.0012 | 0.1056 ± 0.0014 | 0.2390 ± 0.0046 | 0.6125 ± 0.0050 |
| | BPR | 0.2488 ± 0.0030 | 0.0919 ± 0.0013 | 0.1075 ± 0.0013 | 0.2650 ± 0.0040 | 0.6138 ± 0.0034 |
| | FISM | 0.2568 ± 0.0048 | 0.1033 ± 0.0034 | 0.1178 ± 0.0027 | 0.2754 ± 0.0057 | 0.6521 ± 0.0130 |
| | NeuMF | 0.2293 ± 0.0078 | 0.0848 ± 0.0016 | 0.0987 ± 0.0033 | 0.2463 ± 0.0077 | 0.5847 ± 0.0143 |
| | TNF | **0.2775** ± 0.0008 | **0.1075** ± 0.0022 | **0.1235** ± 0.0013 | **0.3012** ± 0.0023 | **0.6579** ± 0.0019 |
| XING5K5K | UCF | 0.0741 ± 0.0012 | 0.0370 ± 0.0017 | 0.0386 ± 0.0012 | 0.0828 ± 0.0014 | 0.2343 ± 0.0033 |
| | MF | 0.0720 ± 0.0026 | 0.0301 ± 0.0013 | 0.0346 ± 0.0018 | 0.0773 ± 0.0026 | 0.2247 ± 0.0086 |
| | BPR | 0.0674 ± 0.0022 | 0.0256 ± 0.0017 | 0.0306 ± 0.0017 | 0.0714 ± 0.0030 | 0.2025 ± 0.0060 |
| | FISM | 0.0835 ± 0.0022 | 0.0379 ± 0.0009 | 0.0427 ± 0.0013 | 0.0898 ± 0.0022 | 0.2648 ± 0.0095 |
| | NeuMF | 0.0481 ± 0.0024 | 0.0166 ± 0.0006 | 0.0195 ± 0.0009 | 0.0507 ± 0.0033 | 0.1347 ± 0.0042 |
| | TNF | **0.0869** ± 0.0017 | **0.0407** ± 0.0012 | **0.0447** ± 0.0008 | **0.0960** ± 0.0027 | **0.2689** ± 0.0070 |

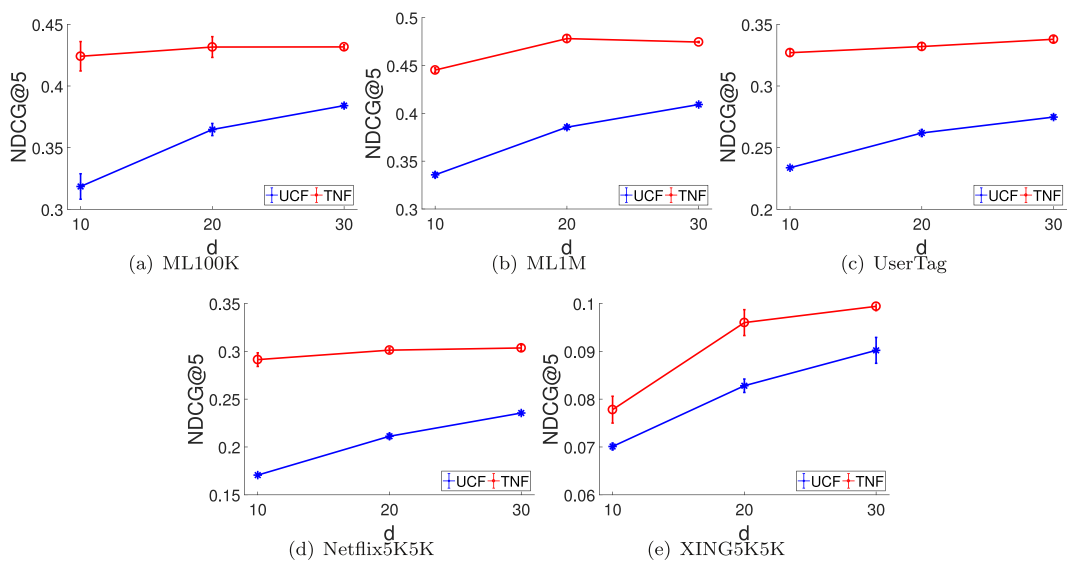



**Fig. 2.** Recommendation performance of user-oriented collaborative filtering (UCF) [8] and our transfer by neighborhood-enhanced factorization (TNF) on five real-world datasets using different neighborhood sizes.

For UCF, we set the size of neighborhood as 20. For BPR, FISM and our TNF, we adopt the commonly used stochastic gradient descent (SGD) method with the same sampling strategy for fair comparison. We fix the dimension as $d = 20$ and learning rate $\gamma = 0.01$. For FISM and TNF, we set $\rho = 3$ [11] and randomly sample $\mathcal{A} = 3|\mathcal{P}|$ (user, item) pairs not included in the observed data $\mathcal{P}$. For the deep model NeuMF, we implement the method using TensorFlow[5] and keep the structure with the best performance as reported in [26], which contains four layers, and validate the performance under the instructions in the original paper. For each factorization-based algorithm on each dataset, we search the trade-off parameters from {0.1, 0.01, 0.001} and find an optimal iteration number from $\{10, 20, 30, \ldots, 990, 1000\}$ by checking the performance of NDCG@5 on the validation data in every ten iterations. Notice that we report the averaged results on three copies of data.

### 4.3. Results

We report the main results in Table 3. We can have the following observations:

[5] https://www.tensorflow.org/.

- TNF performs significantly better than all the five baselines on all the five evaluation metrics across the five datasets, which clearly shows the effectiveness of our transfer learning solution.
- TNF performs much better than the neighborhood-based method, i.e., UCF, in all cases, which showcases the effectiveness of the second task of global preference learning in our TNF.
- TNF is considerably better than the typical globalized factorization-based methods, i.e., MF and BPR, in terms of all evaluation metrics, indicating that it is more effective to learn global preferences via leveraging the local knowledge transferred from the first task of neighborhood construction.
- TNF beats the four very strong baseline methods, i.e., MF, BPR, FISM and NeuMF, in all cases, which showcases the merit of our proposed solution in exploiting the complementarity of the neighborhood-based method and the factorization-based method in a unified framework. In particular, TNF performs significantly better than FISM, which shows the usefulness of the local knowledge as exploited in the second task in TNF.

We further study the impact of the neighborhood sizes in order to compare UCF and our proposed TNF more thoroughly. The results of these two methods using 10, 20 and 30 neighbors are shown in Fig. 2. According to the overall change displayed on five datasets, we can see that the results are relatively stable regardless of different numbers of neighbors, and configuring it as 20 in TNF usually produces the best performance.

## 5. Conclusions and future work

In this paper, we study an important collaborative filtering problem with users' one-class feedback, and design a novel transfer learning solution called *transfer by neighborhood-enhanced factorization* (TNF). In our TNF, the local knowledge of neighborhood among users are extracted from the users' behaviors, which are then transferred to a factorization-based global preference learning task in order to capture the latent representation of users and items better. In this way, our TNF unifies the main idea of neighborhood-based methods and factorization-based methods in a principled way. Experimental results on five real-world datasets show that our TNF can recommend items more accurately than the state-of-the-art methods with regards to various ranking-oriented evaluation metrics.

For future works, we are interested in studying the complementarity of the knowledge of neighborhood and that of the historically observed items, and in incorporating the mined knowledge into deep learning frameworks such as stacked denoising autoencoder and multi-layer neural networks.

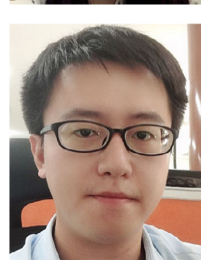
## Acknowledgment

We thank the handling editors and reviewers for their effort and constructive expert comments, and the support of National Natural Science Foundation of China Nos. 61872249, 61502307, 61836005 and 61672358. Weike Pan, Xiaogang Peng and Zhong Ming are corresponding authors for this work.

## References

[1] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the WWW, 2001, pp. 285–295.
[2] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010.
[3] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the ICDM, 2008, pp. 263–272.
[4] R. Pan, Y. Zhou, B. Cao, N.N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: Proceedings of the ICDM, 2008, pp. 502–511.
[5] X. Ning, C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: Recommender Systems Handbook, Springer, 2015, pp. 37–76.
[6] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Proceedings of the ICML, 1998, pp. 46–54.
[7] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[8] C.C. Aggarwal, J.L. Wolf, K.-L. Wu, P.S. Yu, Horting hatches an egg: A new graph-theoretic approach to collaborative filtering, in: Proceedings of the KDD, 1999, pp. 201–212.
[9] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Trans. Inf. Syst. 22 (1) (2004) 143–177.
[10] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: Proceedings of the UAI, 2009, pp. 452–461.
[11] S. Kabbur, X. Ning, G. Karypis, Fism: Factored item similarity models for top-n recommender systems, in: Proceedings of the KDD, 2013, pp. 659–667.
[12] X. He, H. Zhang, M.-Y. Kan, T.-S. Chua, Fast matrix factorization for online recommendation with implicit feedback, in: Proceedings of the SIGIR, 2016, pp. 549–558.
[13] H.-J. Xue, X.-Y. Dai, J. Zhang, S. Huang, J. Chen, Deep matrix factorization models for recommender systems, in: Proceedings of the IJCAI, 2017, pp. 3203–3209.
[14] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[15] J. Wang, A.P. de Vries, M.J.T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, in: Proceedings of the SIGIR, 2006, pp. 501–508.
[16] C.C. Johnson, Logistic Matrix Factorization for Implicit Feedback Data, 27, NIPS, 2014.
[17] Y. Shi, M. Larson, A. Hanjalic, List-wise learning to rank with matrix factorization for collaborative filtering, in: Proceedings of the RecSys, 2010, pp. 269–272.
[18] T. Man, H. Shen, X. Jin, X. Cheng, Cross-domain recommendation: an embedding and mapping approach, in: Proceedings of the IJCAI, 2017, pp. 2464–2470.
[19] B. Li, X. Zhu, R. Li, C. Zhang, Rating knowledge sharing in cross-domain collaborative filtering, IEEE Trans. Cybern. 45 (5) (2015) 1068–1082.
[20] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[21] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NIPS, 2012, pp. 1097–1105.
[22] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the KDD, 2015, pp. 1235–1244.
[23] A.v.d. Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Proceedings of the NIPS, 2013, pp. 2643–2651.
[24] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: a survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) 5:1–5:38.
[25] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the WSDM, 2016, pp. 153–162.
[26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the WWW, 2017, pp. 173–182.
[27] G. Guo, J. Zhang, F. Zhu, X. Wang, Factored similarity models with social trust for top-n item recommendation, Knowl. Based Syst. 122 (C) (2017) 17–25.
[28] W. Pan, L. Chen, Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering, in: Proceedings of the IJCAI, 2013, pp. 2691–2697.
[29] W. Pan, L. Chen, Z. Ming, Personalized recommendation with implicit feedback via learning pairwise preferences over item-sets, Knowl. Inf. Syst. 58 (2) (2019) 295–318.

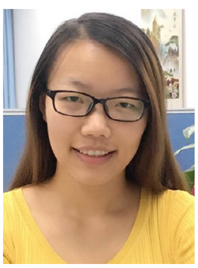
**Wanling Cai** received the B.S. degree in Software Engineering from the Shenzhen University, Shenzhen, China, in 2018. She is currently a Ph.D. student in the department of Computer Science, Hong Kong Baptist University, Hong Kong. Her research interests include data mining and artificial intelligence, mainly focusing on intelligent recommender systems and machine learning. Contact her at cswlcai@comp.hkbu.edu.hk.



**Jiongbin Zheng** received the M.S. degree in Software Engineering from the Shenzhen University, Shenzhen, China, in 2018. He is currently an project manager of a local company in Shenzhen, China. His research interests include intelligent recommendation, transfer learning and machine learning. Contact him at jiongbin92@gmail.com.

**Weike Pan** 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. He is currently an associate professor with the National Engineering Laboratory for Big Data System Computing Technology and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include transfer learning, recommender systems and machine learning. He has been active in professional services. He has served as an editorial board member of Neurocomputing, a co-guest editor of a special issue on big data of IEEE Intelligent Systems (2015-2016), an information officer of ACM Transactions on Intelligent Systems and Technology (2009-2015), and journal reviewer and conference/workshop PC member for dozens of journals, conferences and workshops. Contact him at panweike@szu.edu.cn.

**Jing Lin** received the B.S. degree in Electronic and Information Engineering from the Shenzhen University, Shenzhen, China, in 2018. She is currently a master student in the National Engineering Laboratory for Big Data System Computing Technology and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her research interests include collaborative recommendation, deep learning and transfer learning. Contact her at linjing4@email.szu.edu.cn.



**Lin Li** received the B.S. degree in Computer Science and Technology from the Shenzhen University, Shenzhen, China, in 2017. She is currently a master student in the National Engineering Laboratory for Big Data System Computing Technology and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her research interests include collaborative recommendation, deep learning and transfer learning. Contact her at lilin20171@email.szu.edu.cn.

**Li Chen** is an associate professor at Hong Kong Baptist University. She obtained her Ph.D. degree in human computer interaction at Swiss Federal Institute of Technology in Lausanne (EPFL), and Bachelor and Master degrees in computer science at Peking University, China. Her research interests are mainly in the areas of human-computer interaction, user-centered development of recommender systems and e-commerce decision supports. She is now ACM senior member, and editorial board member of User Modeling and User-Adapted Interaction Journal (UMUAI). She has also been serving in a number of journals and conferences as guest editor, co-organizer, (senior) PC member and reviewer. Contact her at lichen@comp.hkbu.edu.hk.

**Xiaogang Peng** received the Ph.D. degree in Computer-Aided Modeling from the Louisiana Tech University, Louisiana, USA, in 2005. He is currently an associate professor with the National Engineering Laboratory for Big Data System Computing Technology and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include data mining, text information processing and big data. Contact him at pengxg@szu.edu.cn.

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


Please cite this article as: W. Cai, J. Zheng and W. Pan et al., Neighborhood-enhanced transfer learning for one-class collaborative filtering, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.03.016