Knowledge-Based Systems 73 (2015) 173–180

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks

Weike Pan[a], Hao Zhong[b], Congfu Xu[b,*], Zhong Ming[a]

[a] College of Computer Science and Software Engineering, Shenzhen University, China
[b] Institute of Artificial Intelligence, College of Computer Science, Zhejiang University, China

ARTICLE INFO

Article history:
Received 12 April 2014
Received in revised form 28 September 2014
Accepted 28 September 2014
Available online 12 October 2014

Keywords:
Preference learning
Collaborative filtering
Heterogeneous implicit feedbacks
Adaptive Bayesian personalized ranking
Transfer learning

ABSTRACT

Implicit feedbacks have recently received much attention in recommendation communities due to their close relationship with real industry problem settings. However, most works only exploit users' homogeneous implicit feedbacks such as users' *transaction* records from "bought" activities, and ignore the other type of implicit feedbacks like *examination* records from "browsed" activities. The latter are usually more abundant though they are associated with high uncertainty w.r.t. users' true preferences. In this paper, we study a new recommendation problem called heterogeneous implicit feedbacks (HIF), where the fundamental challenge is the uncertainty of the examination records. As a response, we design a novel preference learning algorithm to learn a confidence for each uncertain examination record with the help of transaction records. Specifically, we generalize Bayesian personalized ranking (BPR), a seminal pairwise learning algorithm for homogeneous implicit feedbacks, and learn the confidence adaptively, which is thus called *adaptive Bayesian personalized ranking* (ABPR). ABPR has the merits of uncertainty reduction on examination records and accurate pairwise preference learning on implicit feedbacks. Experimental results on two public data sets show that ABPR is able to leverage uncertain examination records effectively, and can achieve better recommendation performance than the state-of-the-art algorithm on various ranking-oriented evaluation metrics.

1. Introduction

Intelligent recommendations have been widely deployed in various online systems [4,12,28] and mobile applications [14]. Collaborative filtering [6,11,24] as one of the most successful recommendation techniques has been well studied to exploit users' explicit feedbacks such as 5-star graded ratings, especially in the context of Netflix $1 million prize. Most recently, some research works have switched from designing more accurate rating prediction algorithms for explicit feedbacks to developing novel ranking-oriented algorithms for implicit feedbacks [9,17,25], since implicit feedbacks such as users' transaction records are usually more closely related with real industry problem settings.

However, most algorithms for implicit feedbacks only consider one type of data such as users' transaction records. In a real recommendation system, there are usually at least two types of implicit feedbacks [10,16], e.g., users' transaction records and examination records. Note that we use *transaction* and *examination* as an illustrative example, which can be replaced by "bought" (or "watched") and "browsed" in an online e-commerce (or video) system. The implicit feedbacks can also be extended to include more than two types of feedbacks if available. We call this problem *heterogeneous implicit feedbacks* (HIF), which is a natural extension of homogeneous implicit feedbacks studied in [17,25]. In this paper, we focus on this new recommendation problem of HIF.

Different implicit feedbacks in a system are often related though they are different. A (user, item) pair of transaction record usually means that a user likes an item, while a (user, item) pair of examination record from "browsed" activity is of high uncertainty w.r.t. the user's true preference. The fundamental challenge is thus the uncertainty of users' preferences of examination records. Hence, simply combining these two types of feedbacks without distinction may not be the best, which is also supported by our empirical studies.

Can we exploit heterogeneous implicit feedbacks in a principled way? We tackle this new problem from a transfer learning perspective [18], where we take users' transaction records as certain data and users' examination records as uncertain data. To address the uncertainty challenge of examination records, we propose to learn a confidence for each examination record. Specifically, we

[*] Corresponding author.
E-mail addresses: panweike@szu.edu.cn (W. Pan), haozhong@zju.edu.cn (H. Zhong), xucongfu@zju.edu.cn (C. Xu), mingz@szu.edu.cn (Z. Ming).

http://dx.doi.org/10.1016/j.knosys.2014.09.013
0950-7051/© 2014 Elsevier B.V. All rights reserved.

generalize the Bayesian personalized ranking algorithm [25] for homogeneous implicit feedbacks, and design a novel algorithm called *adaptive Bayesian personalized ranking* (ABPR). Our ABPR mainly has two merits, (1) it digests the implicit feedbacks accurately in a pairwise preference learning manner and (2) it learns a confidence for each uncertain feedback in an adaptive manner. Experimental results on two public data sets show that our ABPR is very effective in leveraging uncertain implicit feedbacks, as compared with the state-of-the-art algorithm.

We summarize our main contributions as follows: (1) we study a new recommendation problem called heterogeneous implicit feedbacks (HIF); (2) we design a novel preference learning algorithm called ABPR to fully exploit heterogeneous implicit feedbacks with different uncertainties in a principled way; and (3) we conduct extensive empirical studies and show that our algorithm can produce very promising recommendation results in comparison with the state-of-the-art algorithm.

2. Background

2.1. Problem definition

In our studied problem, there are n users and m items, for which we have two types of implicit feedbacks with different uncertainties. The first type of implicit feedbacks are (user, item) *transaction* records, and the second type are (user, item) *examination* records, which are denoted as $\mathcal{T} = \{(u,i)\}$ and $\mathcal{E} = \{(u,i)\}$, respectively. We illustrate the problem setting using matrix representations in Fig. 1. Our goal is then to fully exploit both data to accurately recommend items to each user.

We list some notations used in the paper in Table 1.

2.2. Bayesian personalized ranking

Bayesian personalized ranking (BPR) [25] is the state-of-the-art algorithm for homogeneous implicit feedbacks, which is based on the assumption that a user prefers a consumed item to an unconsumed item, denoted as $(u,i) \succ (u,j)$ or $\hat{r}_{uij} > 0$. Mathematically, BPR solves the following minimization problem [25],

$$\min_{\Theta} \sum_{(u,i,j):(u,i)\succ(u,j)} f_{uij}(\Theta) + \mathcal{R}_{uij}(\Theta), \tag{1}$$

where $f_{uij}(\Theta) = -\ln\sigma(\hat{r}_{uij})$ is the loss function designed to encourage pairwise competition with $\sigma(x) = 1/(1+\exp(-x))$ and $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$. Note that $\mathcal{R}_{uij}(\Theta) = \frac{\alpha}{2}\|U_{u\cdot}\|^2 + \frac{\alpha}{2}(\|V_{i\cdot}\|^2 + \|V_{j\cdot}\|^2) + \frac{\alpha}{2}(\|b_i\|^2 + \|b_j\|^2)$ is the regularization term used to avoid overfitting, and $\hat{r}_{ui} = \langle U_{u\cdot}, V_{i\cdot}\rangle + b_i$ is the prediction rule based on user u's latent feature vector $U_{u\cdot} \in \mathbb{R}^{1\times d}$, item i's latent feature vector $V_{i\cdot} \in \mathbb{R}^{1\times d}$ and item bias $b_i \in \mathbb{R}$.

Table 1
Some notations.

Notation	Description
$\mathcal{T} = \{(u,i)\}$	Transaction records
$\mathcal{E} = \{(u,i)\}$	Examination records
$(u,i,j)_{\mathcal{T}}$	A triple with $(u,i) \in \mathcal{T},\ (u,j) \notin \mathcal{T}$
$(u,i,j)_{\mathcal{E}}$	A triple with $(u,i) \in \mathcal{E},\ (u,j) \notin \mathcal{T} \cup \mathcal{E}$
$\hat{r}_{ui}$	Preference of user u on item i
$\hat{r}_{uij}$	Preference difference $\hat{r}_{ui} - \hat{r}_{uj}$
$\mathcal{C} = \{c_{ui}\}$	Confidence on examination records $(u,i) \in \mathcal{E}$
Θ	Set of model parameters

The BPR algorithm is a seminal work for homogeneous implicit feedbacks, which has been empirically proved to be very effective [23]. However, it cannot handle the heterogeneity of implicit feedbacks in our studied problem. In the following sections, we will show how we generalize BPR in order to tackle the heterogeneous implicit feedbacks (HIF) problem shown in Fig. 1.

2.3. Bayesian personalized ranking with confidence

BPR with confidence (BPRC) [32] goes one step beyond BPR and include a confidence weight for each implicit feedback [32],

$$\min_{\Theta} \sum_{(u,i,j):(u,i)\succ(u,j)} f_{uij}(c_{uij},\Theta) + \mathcal{R}_{uij}(\Theta), \tag{2}$$

where $f_{uij}(c_{uij},\Theta) = -\ln\sigma(c_{uij}\hat{r}_{uij})$ is a confidence-weighted loss function. We can see that the difference between BPRC in Eq. (2) and BPR in Eq. (1) is the confidence c_{uij} embedded in BPRC. With the given confidence c_{uij}, we can then learn the model parameters in a widely used stochastic gradient descent (SGD) algorithmic framework [11,25],

$$\theta = \theta - \gamma\nabla\theta \tag{3}$$

where γ is the learning rate, θ can be $U_{u\cdot}$, $V_{i\cdot}$, $V_{j\cdot}$, b_i or b_j, and $\nabla\theta$ is the gradient w.r.t. $f_{uij}(c_{uij},\Theta) + \mathcal{R}_{uij}(\Theta)$,

$$\begin{aligned}
\nabla U_{u\cdot} &= -\sigma(-c_{uij}\hat{r}_{uij})(V_{i\cdot} - V_{j\cdot})c_{uij} + \alpha U_{u\cdot},\\
\nabla V_{i\cdot} &= -\sigma(-c_{uij}\hat{r}_{uij})U_{u\cdot}c_{uij} + \alpha V_{i\cdot},\\
\nabla V_{j\cdot} &= -\sigma(-c_{uij}\hat{r}_{uij})(-U_{u\cdot})c_{uij} + \alpha V_{j\cdot},\\
\nabla b_i &= -\sigma(-c_{uij}\hat{r}_{uij})c_{uij} + \alpha b_i,\\
\nabla b_j &= -\sigma(-c_{uij}\hat{r}_{uij})(-1)c_{uij} + \alpha b_j.
\end{aligned}$$

BPRC works well when the confidence for each implicit feedback is given such as that from external context information [32]. However, in most applications such as our studied problem

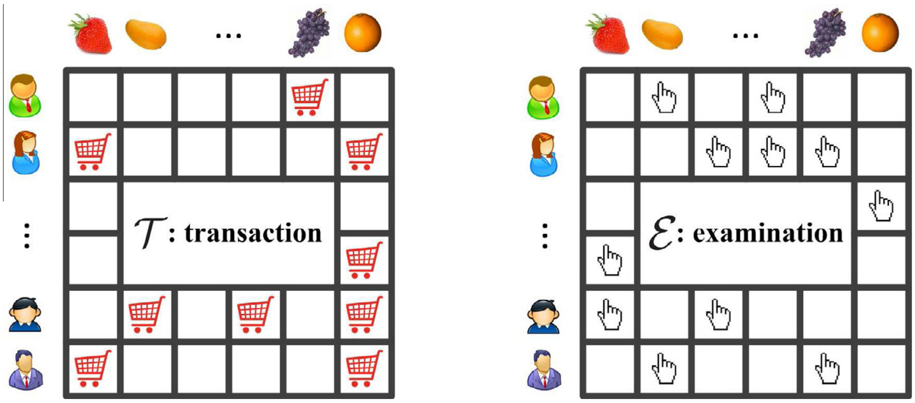


Fig. 1. Illustration of *heterogeneous implicit feedbacks*.

shown in Fig. 1, the confidence may not be available or cannot be easily obtained, which motivates us to learn the confidence.

3. Adaptive Bayesian personalized ranking

When the transaction records $\mathcal{T}$ are few (a.k.a., the transaction data is sparse), BPR [25] may not learn users' preferences well without sufficient training data. The question we ask in this paper is that whether we can leverage some examination records in $\mathcal{E}$ to help reduce the sparsity problem of the transaction data. In order to integrate two different implicit feedbacks in a principled way, the fundamental challenge is the uncertainty associated with the examination records, since a user's examination activity may not necessarily represent a user's "like" or "dislike" preference. The main idea of our solution is to learn a confidence weight for each examination record, rather than to require some external confidence values as that in BPRC [32]. The learned confidence denotes a probability that the corresponding user likes the examined item.

3.1. Objective function

In order to learn model parameters in the preference prediction rule and confidence weight of uncertain records simultaneously, we propose a unified learning framework,

$$\min_{\Theta,\mathcal{C}} \sum_{(u,i,j)} f^{\mathcal{T}}_{uij}(c_{uij},\Theta) + \lambda_{\mathcal{E}} f^{\mathcal{E}}_{uij}(c_{uij},\Theta) + \mathcal{R}_{uij}(\Theta) \tag{4}$$

where (u,i,j) can be $(u,i,j)_{\mathcal{T}}$ and $(u,i,j)_{\mathcal{E}}$ denoting a triple from $\{(u,i,j)|(u,i)\in\mathcal{T},(u,j)\notin\mathcal{T}\}$ and $\{(u,i,j)|(u,i)\in\mathcal{E},(u,j)\notin\mathcal{T}\cup\mathcal{E}\}$, respectively. The difference between our proposed solution in Eq. (4) and BPRC in Eq. (2) is that the confidence values are learned rather than given. Note that we will use c_{ui} to replace the confidence parameter c_{uij} in Eq. (4) since we focus on the uncertainty or confidence of examination records only. We keep the confidence of each transaction record as 1, and use $\mathcal{C}=\{c_{ui}|(u,i)\in\mathcal{E}\}$ to denote the learned confidence of examination records.

From the objective function in Eq. (4), we can see that we have two major terms, $f^{\mathcal{T}}_{uij}(c_{uij},\Theta)$ for transaction records and $f^{\mathcal{E}}_{uij}(c_{uij},\Theta)$ for examination records. Note that $\lambda_{\mathcal{E}}$ is the overall weight assigned to the examination records, which represents how much the uncertain implicit feedbacks will affect the target learning task. In the following sections, we will show how we learn Θ and $\mathcal{C}$ in Eq. (4) via update rules in the widely adopted stochastic gradient descent (SGD) algorithmic framework [11,25].

3.2. Step 1: Learn Θ

In this step, we are given the confidence $\mathcal{C}$, and would like to learn the model parameters in Θ. For a random triple $(u,i,j)_{\mathcal{T}}$, we have the same update rule as Eq. (3). For a random triple $(u,i,j)_{\mathcal{E}}$, we have

$$\theta \leftarrow \theta - \gamma\lambda_{\mathcal{E}}\nabla\theta, \tag{5}$$

where the parameter θ is the same as that in BPRC.

A formal description of the above learning process is given in lines 2–7 in Fig. 2, where K is the inner iteration number used to learn the model parameters sufficiently.

3.3. Step 2: Learn $\mathcal{C}$

In this step, we are given the model parameters of the preference prediction rule Θ, and would like to update the confidence contained in $\mathcal{C}$. We define a binary error function with a threshold τ to reduce the pairwise preference learning problem to a classification problem. We use absolute number (i.e., τ) without normalization because a typical user usually prefers a certain number of items in a recommendation system, which is usually independent of the total number of items. The empirical error is defined as follows,

$$\ell(\hat{r}_{ui}) = \begin{cases} 0, & \text{if } \sum_{j\in\bar{\mathcal{I}}_u}\delta(\hat{r}_{uij}<0)\leqslant\tau, \\ 1, & \text{if } \sum_{j\in\bar{\mathcal{I}}_u}\delta(\hat{r}_{uij}<0)>\tau, \end{cases} \tag{6}$$

where $(u,i)\in\mathcal{E}$, $\bar{\mathcal{I}}_u=\mathcal{I}\setminus\{i|(u,i)\in\mathcal{T}\cup\mathcal{E}\}$, and $\delta(\cdot)$ is an indicator function. We call (u,i) a consistent record if $\ell(\hat{r}_{ui})=0$, and an inconsistent record if $\ell(\hat{r}_{ui})=1$. Note that we define the error on each single (user, item) pair instead of all explicit ratings associated with a certain item in [15], since we aim to learn a confidence for each uncertain examination record with the purpose of leveraging examination records rather than items.

We assume that a user is likely to prefer an examined item to an unexamined item with a similar spirit of pairwise preference learning in BPR [25]. Hence, we consider it a potential error if $\hat{r}_{uij}<0$ for a triple $(u,i,j)_{\mathcal{E}}$. We tolerate the error with a threshold τ and separate it into two values which will decide whether to decrease the corresponding confidence. The error function is important for our confidence update rule and the threshold τ will filter the inconsistent examination records. The value of τ will also decide the number of examination records to be leveraged to the target learning task.

Note that in our confidence update rule, the confidence of each examination record remains unchanged while the confidence of inconsistent records are decreased. Using the empirical error in Eq. (6), we follow the weight update rule in [3],

$$c^{(t)}_{ui} = c^{(t-1)}_{ui}\left(\frac{1}{1+\sqrt{2\ln|\mathcal{E}|/T}}\right)^{\ell\left(\hat{r}^{(t-1)}_{ui}\right)} \tag{7}$$

where T is the number of iterations. Some theoretical analysis has shown good convergence property of this update rule in applications like document classification [3,5]. Specifically, the confidence of a (user, item) examination record will either be fixed, i.e., $c^{(t)}_{ui}=c^{(t-1)}_{ui}$ when $\ell\left(\hat{r}^{(t-1)}_{ui}\right)=0$, or be reduced, i.e., $c^{(t)}_{ui}<c^{(t-1)}_{ui}$ when $\ell\left(\hat{r}^{(t-1)}_{ui}\right)=1$. And after several iterations, an examination record that is consistent with the transaction records will have a large confidence value, while an inconsistent one will have a small confidence value.

A formal description of the above learning process is given in lines 8–10 in Fig. 2.

3.4. The complete algorithm

In our algorithm, we repeat step 1 and step 2 for T times. Each time we generate a base model and calculate the corresponding coefficient. We use the same rule as that in [5] to calculate the coefficient,

$$\beta_t = \frac{1}{2}\ln\left|\left\{(u,i)|\ell\left(\hat{r}^{(t)}_{ui}\right)=0\right\}\right|/\left|\left\{(u,i)|\ell\left(\hat{r}^{(t)}_{ui}\right)=1\right\}\right|, \tag{8}$$

where we can see that the model with more mispredicted records on transaction data will have a lower value of β_t and thus have a smaller impact on the final prediction model.

For the tradeoff parameter $\lambda_{\mathcal{E}}$, it is first initialized to 0 and is then increased gradually to 1 in the iterative process. We may thus regard $\lambda_{\mathcal{E}}$ as an additional overall confidence for examination data. In the beginning, we set equal confidence for examination records, i.e., $c_{ui}=1,\ (u,i)\in\mathcal{E}$. In the iterative process, the confidence of examination records are updated using the empirical error of the previously learned model, and the overall confidence $\lambda_{\mathcal{E}}$ is gradually increased because each examination record is associated with a learned confidence.

Input: transaction records $\mathcal{T} = \{(u,i)\}$, examination records $\mathcal{E} = \{(u,i)\}$, iteration numbers T, K, threshold τ, and base model number T_0.

Output: predicted preference of user u on item i $\hat{y}_{ui}^{(T)} = \sum_{t=T-T_0+1}^{T} \beta_t \hat{r}_{ui}^{(t)}$, learned confidence on each examination record $c_{ui}^{(t)}, (u,i) \in \mathcal{E}$, and coefficient of each base model $\beta_t, t = 1, \ldots, T$.

```
for $t = 1, \ldots, T$ do
    Initialize $U_{u\cdot}, V_{i\cdot}, b_i, 1 \le u \le n, 1 \le i \le m$
    for $k = 1, \ldots, K$ do
        Randomly draw a triple $(u,i,j)_{\mathcal{T}}$
        Update model parameters via Eq.(3)
        Randomly draw a triple $(u,i,j)_{\mathcal{E}}$
        Update model parameters via Eq.(5)
    Calculate the preference $\hat{r}_{ui}^{(t)} = \langle U_{u\cdot}, V_{i\cdot} \rangle + b_i$
    Calculate the empirical error $\ell(\hat{r}_{ui})$ via Eq.(6)
    Update confidence $c_{ui}^{(t)}$ via Eq.(7)
    Calculate the coefficient $\beta_t$ via Eq.(8)
    Update $\lambda_{\mathcal{E}} \leftarrow \lambda_{\mathcal{E}} + 1/(T-1)$
```

Fig. 2. The algorithm of *adaptive Bayesian personalized ranking* (ABPR).

Finally, we can use the most recent T_0 base models to obtain a final preference prediction rule, $\hat{y}_{ui}^{(T)} = \sum_{t=T-T_0+1}^{T} \beta_t \hat{r}_{ui}^{(t)}$, where $\hat{r}_{ui}^{(t)}$ is the predicted preference of user u on item i in the tth base model.

The complete algorithm is shown in Fig. 2. The time complexity of our ABPR is $O(TKd)$, where T is usually a small constant, e.g., $T = 10$ in our empirical studies. Note that BPR [25] is a special case of ABPR with $T = 1$, and its time complexity is then $O(Kd)$. We can thus see that our ABPR algorithm is comparable with the efficient BPR algorithm regarding the time complexity.

4. Experiments

4.1. Data sets

Heterogeneous implicit feedbacks are very common in real industry recommendation systems. However, as far as we know, there is no such public data set freely available. In our empirical studies, we use two real-world data sets, MovieLens[1] and Netflix[2], to simulate the transaction records and examination records.

Both MovieLens and Netflix are users' 5-star ratings on movies, i.e., (user, item, rating) triples. For MovieLens, we randomly take 50% ratings as training data and the remaining 50% ratings as test data. In the training data, we further randomly pick 50% ratings and take the (user, item) pairs with ratings equal to 5 as the transaction records $\mathcal{T}$, in order to simulate "like" preferences. The (user, item) pairs in the remaining 50% data in the training data are used as examination records $\mathcal{E}$, in order to simulate "browsed" activities. In the test data, we adopt the same way as that for training transaction data, and take the (user, item) pairs with ratings equal to 5 as transaction records. For Netflix, we randomly pick 5000 users and 5000 items as a subset in our experiments. For the subset of Netflix, we use the same rule as for MovieLens to construct the transaction records $\mathcal{T}$, examination records $\mathcal{E}$ and test data. For both data sets, we repeat the above procedure for 3 times to generate 3 copies of transaction records, examination records and test data. In our experiments, we report the average recommendation performance and the corresponding standard deviations on those 3 copies of data. The statistics of one copy of the data of MovieLens and Netflix are shown in Table 2.

4.2. Evaluation metrics

Once we have learned the model, we can rank the items based on the estimated preference scores. We use $I_u(p)$ and $P_u(i)$ to denote the item located at ranked position p and the ranked posi-

[1] http://www.grouplens.org/node/73/.

[2] http://www.netflixprize.com/.

Table 2
Statistics of data sets.

	MovieLens	Netflix
User number (n)	6040	5000
Item number (m)	3952	5000
Transaction records ($\|\mathcal{T}\|$)	56,619	19,903
Examination records ($\|\mathcal{E}\|$)	249,994	82,831
Transaction records (test data)	113,269	39,707

tion of item i for user u, respectively. In order to study the empirical performance of our ABPR extensively, we adopt five ranking-oriented metrics, which have been widely used in evaluation of information retrieval and recommendation algorithms, including precision, normalized discounted cumulative gain (NDCG) [33], mean reciprocal rank (MRR) [29], average relative position (ARP) [31] and area under the curve (AUC) [25]. Mathematically, they are defined as follows [19],

- $Pre@k = \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{1}{k}\sum_{p=1}^{k}\delta(I_u(p)\in\mathcal{I}_u^{te})$, where $\mathcal{U}^{te}$ is the set of users in the test data, $\mathcal{I}_u^{te}$ is the set of preferred items by user u in the test data, and $\delta(x)$ is an indicator function with $\delta(x) = 1$ if x is true and $\delta(x) = 0$ otherwise;
- $NDCG@k = \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{1}{Z_u}\sum_{p=1}^{k}\frac{2^{\delta(I_u(p)\in\mathcal{I}_{te}^{u})}-1}{\log(p+1)}$, where Z_u is a normalization term with preferred items ranked first, i.e., $Z_u = \sum_{p=1}^{\min(k,|\mathcal{I}_u^{te}|)}\frac{1}{\log(p+1)}$;
- $MRR = \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{1}{\min_{i\in\mathcal{I}_u^{te}}P_u(i)}$;
- $ARP = \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{1}{|\mathcal{I}_u^{te}|}\sum_{i\in\mathcal{I}_u^{te}}\frac{P_u(i)}{|\mathcal{I}^{tr}|-|\mathcal{I}_u^{tr}|}$, where $\mathcal{I}^{tr}$ is the set of items in the training data; and
- $AUC = \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{1}{|\mathcal{R}^{te}(u)|}\sum_{(i,j)\in\mathcal{R}^{te}(u)}\delta(\hat{r}_{ui} > \hat{r}_{uj})$, where $\mathcal{R}^{te}(u) = \{(i,j)|(u,i)\in\mathcal{T}^{te},(u,j)\notin\mathcal{T}\cup\mathcal{T}^{te}\}$ with $\mathcal{T}$ and $\mathcal{T}^{te}$ as transaction records in the training data and test data, respectively.

4.3. Baselines and parameter settings

In order to study the effect of the learned confidence more directly, we compare our preference learning algorithm with the state-of-the-art algorithm for implicit feedbacks, Bayesian personalized ranking [25], including BPR for transaction data $\mathcal{T}$ only and BPR($\mathcal{T}\cup\mathcal{E}$) for the combination of transaction data and examination data. Besides BPR, we also use a common method based on items' popularity, i.e., PopRank [29]. For fair comparison, we implement the BPR algorithm and our ABPR algorithm both in Python in the same algorithmic framework (see Fig. 2). The initializations of the model variables are the same as [20]. Note that BPRC [32] is not applicable to our studied problem because the required confidence is not available.

For fair comparison, we adopt the same way of parameter setting for BPR, BPR($\mathcal{T}\cup\mathcal{E}$) and our APBR in the experiments. For the inner iteration number K, we set it to a relatively large value, 3×10^8, to ensure that it reaches sufficient convergence. For the iteration number T, we have tried $T\in\{10,20\}$ for both data sets (with $d = 10$), and found that the performance with $T\in\{10,20\}$ are very similar, which means that our ABPR algorithm converges in a few iterations. Hence, we fix $T = 10$ for all the experiments. For the number of latent features, we have tried $d\in\{10,20\}$ [20]. For the regularization parameter α, we have tried $\alpha = \{0.001, 0.01, 0.1\}$ [20] and picked the one that has the best result w.r.t. the *NDCG*@5 metric on the first copy of data, and then fix them in the rest two copies of data. We find that the best values of the regularization coefficient is 0.01 for both data sets. We fix the learning rate γ as 0.01 [20]. For the empirical error threshold, τ cannot be set too large or too small, and the range of τ may also be different for different data sets. We have tried different values of τ in the experiments and show the details in the following sections. The confidence of each examination record is initialized as 1. In the final prediction rule of our ABPR, we take the most recent three (i.e., $T_0 = 3$) base models.

4.4. Summary of experimental results

We compare the performance of our ABPR algorithm with two baselines: BPR [25] and PopRank [29]. The recommendation performance on five evaluation metrics are shown in Tables 3,4, where $\tau = 600$ for MovieLens and $\tau = 150$ for Netflix (the impact of the parameter τ will be shown in subsequent sections). We can have the following observations:

1. our ABPR algorithm beats all baselines on all evaluation metrics, which clearly shows the effectiveness of our preference learning approach;
2. the overall performance ordering is ABPR > BPR($\mathcal{T}\cup\mathcal{E}$) > BPR > PopRank, which shows that (1) the pairwise preference learning algorithms are effective since PopRank is the worst, and (2) the examination data is useful since both ABPR and BPR($\mathcal{T}\cup\mathcal{E}$) are better than BPR; and
3. ABPR is better than BPR($\mathcal{T}\cup\mathcal{E}$), which shows that the learned confidence in ABPR is helpful in addressing the uncertainty challenge of examination data, since BPR($\mathcal{T}\cup\mathcal{E}$) can be considered as a special case of our ABPR with constant confidence $c_{ui} = 1$.

In order to better understand the effectiveness of our preference learning algorithm for different user groups, we conduct a fine-grained analysis on the recommendation performance. We divide the users of MovieLens and Netflix into 10 and 7 groups, respectively, where users in different groups have different numbers of transaction records. The details of user groups and the corresponding performance are shown in Fig. 3. From the results in Fig. 3, we can have the following observations:

1. the overall trends show that the recommendation performance increases when users are with more feedbacks, which is consistent with various existing works on recommendation algorithms;
2. the performance of BPR ($\mathcal{T}\cup\mathcal{E}$) and ABPR are much better than BPR and PopRank, which again shows that the examination data is useful; and
3. ABPR performs best on most user groups, which shows the effectiveness of our preference learning algorithm in uncertainty reduction, i.e., learning the confidence of each examination record.

We further study the impact of the threshold parameter τ in the error function shown in Eq. (6). As mentioned before, the value of τ cannot be set too large or too small. In other words, τ has a value range (τ_{min}, τ_{max}). In our experiments, we have tried several different values of τ and found that for different data sets and different numbers of latent features d, the maximal value τ_{max} is different. For the consistency with the parameter search as described in Section 4.3, we use the metric *NDCG*@5 to study the effect of the parameter τ and report the results in Fig. 4, from which we can have the following observations:

1. τ_{max} for Netflix is smaller than τ_{max} for MovieLens, which is caused by the fewer pairwise constraints in Netflix since it is sparser than MovieLens;

Table 3
Recommendation performance of ABPR and other algorithms on MovieLens ($d = 20$).

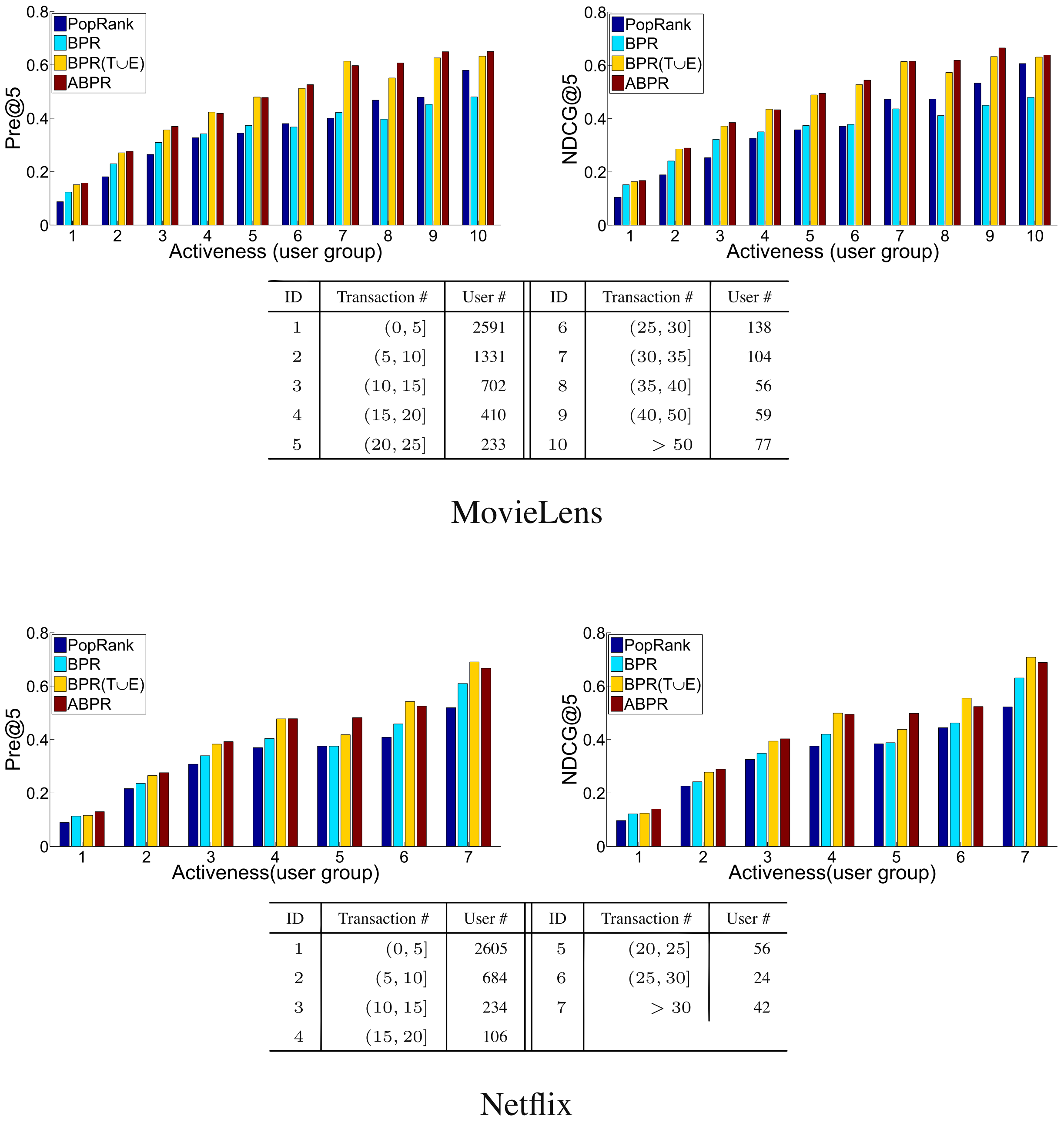
Algorithm	*Pre*@5 ↑	*NDCG*@5 ↑	*MRR* ↑	*ARP* ↓	*AUC* ↑
PopRank	0.1769 ± 0.0021	0.1859 ± 0.0026	0.3361 ± 0.0034	0.0889 ± 0.0010	0.8746 ± 0.0007
BPR	0.2061 ± 0.0019	0.2169 ± 0.0052	0.3845 ± 0.0048	0.0738 ± 0.0009	0.9011 ± 0.0002
BPR($\mathcal{T} \cup \mathcal{E}$)	0.2548 ± 0.0034	0.2654 ± 0.0042	0.4512 ± 0.0047	0.0681 ± 0.0013	0.9098 ± 0.0007
ABPR	**0.2638** ± 0.0045	**0.2781** ± 0.0031	**0.4817** ± 0.0054	**0.0650** ± 0.0018	**0.9142** ± 0.0010

Numbers in boldface (e.g., 0.2638) are the best results.

Table 4
Recommendation performance of ABPR and other algorithms on Netflix ($d = 20$).

Algorithm	*Pre*@5 ↑	*NDCG*@5 ↑	*MRR* ↑	*ARP* ↓	*AUC* ↑
PopRank	0.1169 ± 0.0023	0.1233 ± 0.0025	0.2531 ± 0.0034	0.1183 ± 0.0005	0.8841 ± 0.0004
BPR	0.1315 ± 0.0022	0.1373 ± 0.0023	0.2912 ± 0.0052	0.0669 ± 0.0008	0.9017 ± 0.0004
BPR($\mathcal{T} \cup \mathcal{E}$)	0.1475 ± 0.0022	0.1578 ± 0.0033	0.3345 ± 0.0042	0.0581 ± 0.0014	0.9074 ± 0.0013
ABPR	**0.1569** ± 0.0037	**0.1668** ± 0.0022	**0. 3511** ± 0.0039	**0.0520** ± 0.0015	**0.9123** ± 0.0010

Numbers in boldface (e.g., 0.1569) are the best results.



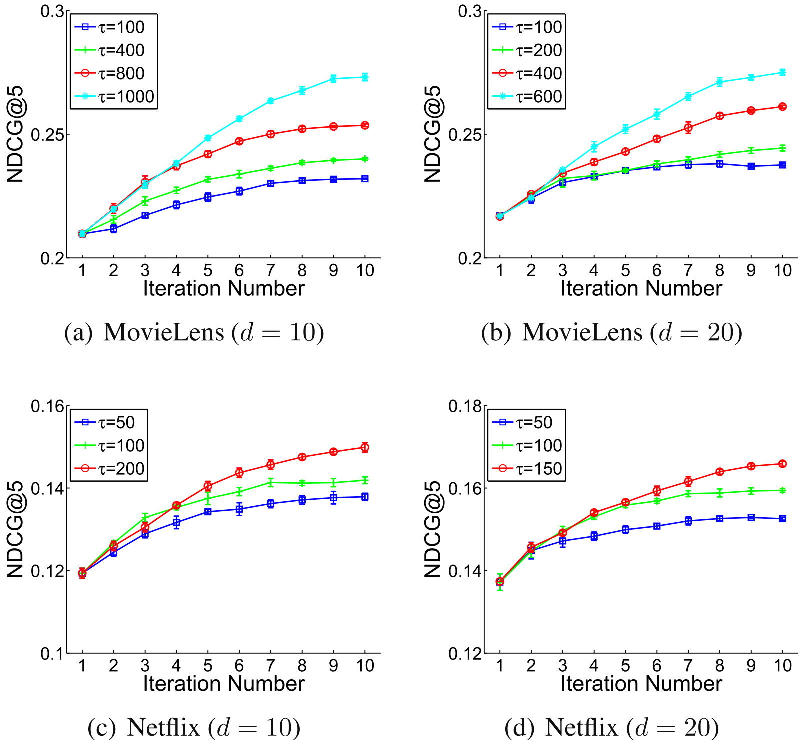
ID	Transaction #	User #	ID	Transaction #	User #
1	(0, 5]	2591	6	(25, 30]	138
2	(5, 10]	1331	7	(30, 35]	104
3	(10, 15]	702	8	(35, 40]	56
4	(15, 20]	410	9	(40, 50]	59
5	(20, 25]	233	10	> 50	77

MovieLens

ID	Transaction #	User #	ID	Transaction #	User #
1	(0, 5]	2605	5	(20, 25]	56
2	(5, 10]	684	6	(25, 30]	24
3	(10, 15]	234	7	> 30	42
4	(15, 20]	106			

Netflix

Fig. 3. Recommendation performance of ABPR and other algorithms on different user groups of MovieLens and Netflix ($d = 20$).



(a) MovieLens ($d = 10$)

(b) MovieLens ($d = 20$)

(c) Netflix ($d = 10$)

(d) Netflix ($d = 20$)

Fig. 4. Recommendation performance of ABPR with different values of τ in Eq. (6).

2. τ_{max} (with $d = 20$) is smaller than τ_{max} (with $d = 10$), which means that a more flexible model (larger value of d) can satisfy the pairwise constraints $\hat{r}_{ui} > \hat{r}_{uj}$ in preference learning more easily; and
3. when $\tau \in (\tau_{min}, \tau_{max})$, a larger value of τ can generate better recommendation performance, which shows that the examination records are useful and the learned confidence are helpful since a larger value of τ means leveraging more confidence-weighted examination records.

5. Related works

Recommendation techniques [1,2,27] usually learn users' preferences from the recorded feedbacks and other available information, in which users' feedbacks are critical for the performance of the personalized services to be provided since they are directly related to users' true preferences. We thus put our work in the context of recommendation with different feedbacks and categorize the related works into homogeneous feedbacks and heterogeneous feedbacks.

Homogeneous feedbacks. Homogeneous feedbacks include explicit feedbacks such as 5-star ratings and implicit feedbacks like "browsed" activities. So far, various collaborative filtering algorithms have been proposed, including, (1) maximum margin matrix factorization (MMMF) [30], probabilistic matrix factorization (PMF) [26], SVD++ [11] and Fuzzy-based Telecom Product Recommender System (FTCP-RS) [34] for homogeneous explicit feedbacks, (2) one-class collaborative filtering (OCCF) [17], implicit matrix factorization (iMF) [8], Bayesian personalized ranking (BPR) [25], factored item similarity models (FISM)[9] and group preference based BPR (GBPR) [20] for homogeneous implicit feedbacks.

Heterogeneous feedbacks. Heterogeneous feedbacks usually refer to a situation with more than one type of users' feedbacks such as 5-star numerical ratings and like/dislike binary scores as that in transfer by collective factorization (TCF) [22], in which transfer learning [18] techniques have played an important role. From the perspective of "what knowledge to transfer" [18,21], our ABPR algorithm takes each examination record as an implicit preference instance, which can thus be considered as an instance-based transfer learning algorithm [3,21]. From the perspective of "how to transfer knowledge", our ABPR algorithm integrates the examination records into a unified preference learning framework with the learned confidence, which is thus an integrative transfer learning algorithm [21].

Table 5
Summary of some related works in recommendation w.r.t. users' feedbacks.

	Explicit feedbacks	Implicit feedbacks
Homogeneous feedbacks	PMF, SVD++, etc.	OCCF, BPR, etc.
Heterogeneous feedbacks	TCF, etc.	ABPR

The difference between our work and the aforementioned works can be identified from two perspectives, (1) we study a new recommendation problem (i.e., heterogeneous implicit feedbacks, HIF) rather than existing problems with known solutions and (2) we propose a novel preference learning algorithm for HIF, which has the merits of accurate pairwise preference learning for implicit feedbacks and adaptive confidence learning for uncertain feedbacks. We summarize some related works w.r.t. users' feedbacks in Table 5, from which we can see that our ABPR is a novel algorithm for a new recommendation problem.

6. Conclusions and future work

In this paper, we study a new recommendation problem called heterogeneous implicit feedbakcs (HIF) shown in Fig. 1, which includes two types of implicit feedbacks, i.e., users' *transaction* records and *examination* records. In order to fully exploit these two types of feedbacks with different uncertainties in a principled way, we propose a novel preference learning algorithm called *adaptive Bayesian personalized ranking* (ABPR). Specifically, ABPR generalizes a seminal work called BPR [25] and learns a confidence for each examination record adaptively so as to address the fundamental challenge of uncertainty. With the learned confidence, the uncertain examination records can be integrated into the target recommendation task in a unified pairwise preference learning framework. Empirically, we have observed very promising recommendation results on two public data sets as compared with the state-of-the-art recommendation algorithm on various ranking-oriented evaluation metrics.

For future work, we are interested in (1) deploying our algorithm in real e-commence settings and (2) designing a general preference learning solution for HIF and social contextual information [7,13].

Acknowledgements

We thank the support of National Natural Science Foundation of China (NSFC) No. 61272303 and National Basic Research Program of China (973 Program) No. 2010CB327903, Natural Science Foundation of SZU No. 201436, NSFC No. 61170077, NSF GD No. 10351806001000000, S&T Project of GDA No. 2012B091100198, and S&T Project of SZ No. JCYJ20130326110956468.

References

[1] Gediminas Adomavicius, Alexander Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. (IEEE TKDE) 17 (6) (2005) 734–749.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. GutiéRrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[3] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, Yong Yu, Boosting for transfer learning, in: Proceedings of the 24th International Conference on Machine Learning, ICML '07, 2007, pp. 193–200.

[4] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, Shyam Rajaram, Google news personalization: scalable online collaborative filtering, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, 2007, pp. 271–280.

[5] Yoav Freund, Robert E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the 2nd European Conference on Computational Learning Theory, EuroCOLT '95, 1995, pp. 23–37.

[6] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry, Using collaborative filtering to weave an information tapestry, Commun. ACM (CACM) 35 (12) (1992) 61–70.

[7] Guibing Guo, Jie Zhang, Daniel Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowl.-Based Syst. 57 (2014) 57–68.

[8] Yifan Hu, Yehuda Koren, Chris Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the 8th IEEE International Conference on Data Mining, ICDM '08, 2008, pp. 263–272.

[9] Santosh Kabbur, Xia Ning, George Karypis, FISM: factored item similarity models for top-N recommender systems, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, 2013, pp. 659–667.

[10] Diane Kelly, Jaime Teevan, Implicit feedback for inferring user preference: a bibliography, SIGIR Forum 37 (2) (2003) 18–28.

[11] Yehuda Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, 2008, pp. 426–434.

[12] Greg Linden, Brent Smith, Jeremy York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (1) (2003) 76–80.

[13] Nathan N. Liu, Luheng He, Min Zhao, Social temporal collaborative ranking for context aware movie recommendation, ACM Trans. Intell. Syst. Technol. (ACM TIST) 4 (1) (2013) 15:1–15:26.

[14] Qi Liu, Haiping Ma, Enhong Chen, Hui Xiong, A survey of context-aware mobile recommendations, Int. J. Inform. Technol. Decis. Mak. 12 (1) (2013) 139–172.

[15] Zhongqi Lu, Weike Pan, Evan Wei Xiang, Qiang Yang, Lili Zhao, ErHeng Zhong, Selective transfer learning for cross domain recommendation, in: Proceedings of SIAM Data Mining, SDM '13, 2013, pp. 641–649.

[16] Douglas Oard, Jinmook Kim, Implicit feedback for recommender systems, in: Proceedings of the AAAI Workshop on Recommender Systems, 1998, pp. 81–83.

[17] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, Qiang Yang, One-class collaborative filtering, in: Proceedings of the 8th IEEE International Conference on Data Mining, ICDM '08, 2008, pp. 502–511.

[18] Sinno Jialin Pan, Qiang Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. (IEEE TKDE) 22 (10) (2010) 1345–1359.

[19] Weike Pan, Li Chen, CoFiSet: collaborative filtering via learning pairwise preferences over item-sets, in: Proceedings of SIAM Data Mining, SDM '13, 2013, pp. 180–188.

[20] Weike Pan, Li Chen, GBPR: group preference based bayesian personalized ranking for one-class collaborative filtering, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13, 2013, pp. 2691–2697.

[21] Weike Pan, Evan W. Xiang, Qiang Yang, Transfer learning in collaborative filtering via uncertain ratings, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI '12, 2012, pp. 662–668.

[22] Weike Pan, Qiang Yang, Transfer learning in heterogeneous collaborative filtering domains, Artif. Intell. 197 (2013) 39–55.

[23] Ulrich Paquet, Noam Koenigstein, One-class collaborative filtering with random graphs, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, 2013, pp. 999–1008.

[24] Steffen Rendle, Factorization machines with libFM, ACM Trans. Intell. Syst. Technol. (ACM TIST) 3 (3) (2012) 57:1–57:22.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI '09, 2009, pp. 452–461.

[26] Ruslan Salakhutdinov, Andriy Mnih, Probabilistic matrix factorization, in: Annual Conference on Neural Information Processing Systems 20, NIPS '08, 2008, pp. 1257–1264.

[27] Qusai Shambour, Jie Lu, A trust-semantic fusion-based recommendation approach for e-business applications, Decis. Support Syst. 54 (1) (2012) 768–780.

[28] Amit Sharma, Baoshi Yan, Pairwise learning in recommendation: experiments with community recommendation on linkedin, in: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, 2013, pp. 193–200.

[29] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, Alan Hanjalic, CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering, in: Proceedings of the 6th ACM Conference on Recommender Systems, RecSys '12, 2012, pp. 139–146.

[30] Nathan Srebro, Jason D.M. Rennie, Tommi Jaakkola, Maximum-margin matrix factorization, in: Annual Conference on Neural Information Processing Systems 16, NIPS '04, 2004, pp. 1329–1336.

[31] Gábor Takács, Domonkos Tikk, Alternating least squares for personalized ranking, in: Proceedings of the 6th ACM Conference on Recommender Systems, RecSys '12, 2012, pp. 83–90.

[32] Sheng Wang, Xiaobo Zhou, Ziqi Wang, Ming Zhang, Please spread: recommending tweets for retweeting with implicit feedback, in: Proceedings of the 2012 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media, DUBMMSM '12, 2012, pp. 19–22.

[33] Shuang-Hong Yang, Bo Long, Alexander J. Smola, Hongyuan Zha, Zhaohui Zheng, Collaborative competitive filtering: learning recommender using context of user choice, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, 2011, pp. 295–304.

[34] Zui Zhang, Hua Lin, Kun Liu, Dianshuang Wu, Guangquan Zhang, Jie Lu, A hybrid fuzzy-based personalized recommender system for telecom products/services, Inform. Sci. 235 (2013) 117–129.