Collaborative Users' Brand Preference Mining across Multiple Domains from Implicit Feedbacks

Jian Tang[1*], Jun Yan[2], Lei Ji[2], Ming Zhang[1], Shaodan Guo[3*], Ning Liu[2], Xianfang Wang[4] and Zheng Chen[2]

[1]Peking University, Beijing, China, {tangjian, mzhang}@net.pku.edu.cn
[2]Microsoft Research Asia, {junyan, leiji, ningl, zhengc }@microsoft.com
[3] Huazhong University of Science and Technology, Wuhan, China, guoshaodan@gmail.com
[4]Microsoft Adcenter Audience Intelligence, Redmond, WA, United States, xfwang@microsoft.com

Abstract

Advanced e-applications require comprehensive knowledge about their users' preferences in order to provide accurate personalized services. In this paper, we propose to learn users' preferences to product brands from their implicit feedbacks such as their searching and browsing behaviors in user Web browsing log data. The user brand preference learning problem is challenge since (1) the users' implicit feedbacks are extremely sparse in various product domains; and (2) we can only observe positive feedbacks from users' behaviors. In this paper, we propose a latent factor model to collaboratively mine users' brand preferences across multiple domains simultaneously. By collective learning, the learning processes in all the domains are mutually enhanced and hence the problem of data scarcity in each single domain can be effectively addressed. On the other hand, we learn our model with an adaption of the Bayesian personalized ranking (BPR) optimization criterion which is a general learning framework for collaborative filtering from implicit feedbacks. Experiments with both synthetic and real world datasets show that our proposed model significantly outperforms the baselines.

Introduction

Mining users' preferences [Holland et al. 2003, Jung et al. 2005] is critical in many applications, such as advanced e-applications, and online advertising. In these applications, we require comprehensive knowledge about users' likes and dislikes in order to provide individual product recommendations or user-specific advertisement delivery. Then the users are more likely to buy the recommended products or click the delivered advertisements, which match users' preferences.

In this paper, we propose to mine users' brand preferences from their implicit feedbacks, such as searching and browsing. If a user searches the product of a brand or visits the homepage of the brand, then we assume the user has positive feedbacks over the brand. Based on the assumption "*similar users tend to have similar taste*", we formulate the task into a collaborative filtering problem. However, different from classical collaborative filtering problem, which learns users' preferences from their explicit feedbacks, our problem suffers from two major challenges:

(1) Extreme data sparsity. Users' implicit feedbacks are quite few in each domain, and this will result that the user-brand matrix in each domain are extremely sparse. We cannot accurately learn users' brand preferences in each domain by only exploiting their behaviors in the domain.

(2) No negative feedbacks. From users' online behaviors, we can only observe their positive feedbacks over the brands, such as searching the product of the brands or visiting the homepages of the brands.

In this paper, we propose a latent factor model to collaboratively mine users' brand preferences across multiple domains simultaneously. Users' behaviors in different domains are correlated. If we find a user prefers famous brands in several domains, then he would also be likely to prefer famous brands in other domains. Thus by collective learning, the learning processes of all the domains are mutually enhanced and the first challenge can be addressed. Besides, our model implicitly models the similarities between domains, which have been proved quite helpful in improving the performance of jointly learning users' preferences across multiple domains from explicit feedbacks [Cao et al. 2010]. For addressing the second challenge, we learn our proposed model with the **Bayesian Personalized Ranking** (**BPR**) optimization

*This work was done when the first and fourth author were visiting Microsoft Research Asia.

criterion [Rendle et al. 2009], which is a general learning framework for collaborative filtering from users' implicit feedbacks. Experiments with both synthetic and real world datasets show that the collective learning methods for multiple domains outperform the methods for single domain and our proposed model performs the best due to the consideration of similarities between domains.

Users' Brand Preference Mining

The task of users' brand preference mining is to provide each user with a ranked list of brands in each domain. In this paper, we consider mining users' brand preferences from their implicit feedbacks, such as searching and browsing. The key difference between users' implicit feedbacks and their explicit feedbacks is that only positive feedbacks are available. Based on the basic assumption "*similar users tend to have similar taste*", we propose to collaboratively mine users' brand preferences, which results in a collaborative filtering problem. Furthermore, users' behaviors in each domain are quite scarce, thus we propose to mine users' brand preferences across multiple domains simultaneously.

Notations

We first define the notations to be used in this paper. Let D be the set of all domains, U the set of all users, and B_d the set of brands in the domain d. We use r_{dui} to represent the number of observations between user u and brand i in domain d, which refers to the number of times user u searched queries containing brand i or visited the homepage of brand i. We introduce another set of binary variables p_{dui} to indicate whether user u likes brand i. The p_{dui} values are defined as follows:

$$p_{dui} = \begin{cases} 1 & \text{if } r_{dui} > 0 \\ 0 & \text{else} \end{cases} \qquad (1)$$

From the definition of p_{dui}, we can see that p_{dui} indicate whether user u has positive feedbacks over brand i, ignoring the number of times. Furthermore, a set of variables c_{dui} are introduced to measure the confidence in observing p_{dui}, defined as:

$$c_{dui} = 1 + \kappa r_{dui} \qquad (2)$$

where κ is the increase rate of confidence value with respect to the number of positive feedbacks. We set $\kappa = 0.5$ in our experiments for the best performance.

Problem Formulation

The problem of users' brand preference mining is to provide a personalized ranked list of brands in each domain. For each domain d and user u, we want to predict a total order $>_{d,u} \subset B_d \times B_d$ over the brands, and meanwhile $>_{d,u}$ has to meet the properties of a total order:

$$\forall i, j: i \neq j \Rightarrow i >_{d,u} j \lor j >_{d,u} i \qquad \text{(totality)}$$
$$\forall i, j: i >_{d,u} j \land j >_{d,u} i \Rightarrow i = j \qquad \text{(antisymmetry)}$$
$$\forall i, j, k: i >_{d,u} j \land j >_{d,u} k \Rightarrow i >_{d,u} k \qquad \text{(transitivity)}$$

For convenience, we define:

$$P_u^d = \{i | i \in B_d: p_{dui} > 0\},$$

that denotes the set of brands for which user u has positive feedbacks in domain d.

Data Analysis

The main problem of learning from users' implicit feedbacks is that only positive feedbacks are available and the remaining data is a mixture of positive and negative examples. Many classical collaborative filtering methods create the training data by giving each $i \in P_u^d$ a positive class label and others a negative one [Hu et al. 2008, Pan et al. 2008], and then a model is fitted to this data. As mentioned in [Rendle et al. 2009], for a model with enough expressiveness, this model cannot rank at all as it predicts only 0s for the remaining data. Here we use a similar strategy as done in [Rendle et al. 2009], we create the brand preference pairs as our training data. If user u has positive feedbacks for brand i but none for brand j, we assume user u prefers i to j. In [Rendle et al. 2009], all the preference pairs are treated equally, but here we associate each preference pair with the confidential value c_{dui}, which measures the confidence of user u prefers i to j. Thus the training data derived from user u's behaviors in domain d is defined as:

$$T_u^d = \{((d, u, i, j), c_{dui}) | i \in P_u^d, j \in B_d \backslash P_u^d\}, \qquad (3)$$

and the total training data T is:

$$T = \cup_{d,u} T_u^d \qquad (4)$$

Bayesian Personalized Ranking (BPR) for Users' Brand Preference Mining

In this section, we present a solution for the problem of users' brand preference mining. First, we describe a learning framework for collaborative filtering from implicit feedbacks: Bayesian Personalized Ranking (BPR) optimization criterion [Rendle et al. 2009]. Second, we describe the factorization models. We first describe the collective matrix factorization model with BPR optimization criterion. Then we describe our factorization model for the problem and present the detailed learning algorithm with BPR optimization criterion.

Bayesian Personalized Ranking Criterion

In this subsection, we describe the **Bayesian Personalized Ranking (BPR)** criterion, which is a general framework for collaborative filtering from implicit feedbacks.

In order to find the best ranking $>_{d,u} \subset B_d \times B_d$, the BPR framework aims to maximize the following posterior probability:

$$p(\Theta| >_{d,u}) \propto p(>_{d,u} |\Theta)p(\Theta) \quad (5)$$

where Θ represents the model parameters. By assuming the independence of users' behaviors in different domains and the independence of users in each domain, the model parameters can be found by:

$$argmax_{\Theta} \prod_{d} \prod_{u} p(>_{d,u} |\Theta)p(\Theta) \quad (6)$$

Furthermore, $p(>_{d,u} |\Theta)$ can be expressed with

$$p(>_{d,u} |\Theta) = \prod_{((d,u,i,j),c_{dui}) \in T_u^d} p(i >_{d,u} j|\Theta)^{c_{dui}} \quad (7)$$

The probability that user *u* prefers brand *i* to *j* is defined as:

$$p(i >_{d,u} j|\Theta) = \sigma\left(\hat{p}_{duij}(\Theta)\right) \quad (8)$$

where σ is the logistic function $\sigma(x) = 1/(1+e^{-x})$. $\hat{p}_{duij}(\Theta)$ is a real-valued function of model parameters Θ, i.e. $\hat{p}: D \times U \times B_d^2 \to R$. We will use $\hat{p}_{duij}$ to represent $\hat{p}_{duij}(\Theta)$ for short in the rest of the paper.

We have discussed the likelihood function. As for the prior $p(\Theta)$, we assume Θ follows a multivariate Gaussian distribution $\Theta \sim N(0, \lambda_{\Theta} I)$. In total, the BPR optimization criterion can be finally formulated as:

$$\begin{aligned} \text{BPR-OPT} &:= \ln p(\Theta| >_{d,u}) \\ &= \ln p(>_{d,u} |\Theta) + \ln p(\Theta) \end{aligned} \quad (9)$$

The optimization criterion (9) is differentiable, thus gradient descent based algorithm can be easily adapted to the problem. Instead, a bootstrap sampling of training data based stochastic gradient-descent algorithm is proposed [Rendle et al. 2009]. Given a training example (d, u, i, j), the gradient of BPR-OPT with respect to a model parameter θ is calculated as:

$$\begin{aligned} &\frac{\partial}{\partial \theta}\left(c_{dui} \ln \sigma(\hat{p}_{duij}(\Theta)) - \lambda_{\Theta} \parallel \Theta \parallel^2\right) \\ &\propto c_{dui}\left(1 - \sigma(\hat{p}_{duij})\right)\frac{\partial}{\partial \theta}\hat{p}_{duij} - \lambda_{\Theta}\theta \end{aligned} \quad (10)$$

And the parameter θ can be updated with:

$$\theta \leftarrow \theta + \alpha\left(c_{dui}\left(1 - \sigma(\hat{p}_{duij})\right)\frac{\partial}{\partial \theta}\hat{p}_{duij} - \lambda_{\Theta}\theta\right) \quad (11)$$

where α is the learning rate.

Factorization Models

Factorization models are quite popular for collaborative filtering problems and have been proved quite successful [Hu et al. 2008, Rendle et al. 2009]. In this subsection, we describe two factorization models for the problem of users' brand preference mining. The first one is the classical matrix factorization method for collaborative filtering problem, and the other one is our proposed model. We will present how the two models are learned with BPR.

Both of the two factorization models predict a real-valued function $\hat{p}: D \times U \times B_d \to R$, where the score $\hat{p}_{dui}$ in entry (d, u, i) represents user *u*'s preference score for brand *i* in domain *d*. In order to learn the models with BPR optimization criterion, we define:

$$\hat{p}_{duij} = \hat{p}_{dui} - \hat{p}_{duj}$$

Collective Matrix Factorization

The matrix factorization methods or latent factor models associate each user *u* with a latent vector $x_u \in R^f$ and brand *i* in domain *d* with a latent vector $y_i^d \in R^f$, where *f* is the dimension of the latent factors. The preference score $\hat{p}_{dui}$ is modeled via:

$$\hat{p}_{dui} = x_u^T y_i^d \quad (12)$$

The model parameters for the matrix factorization are $\Theta = \{x_u, y_i^d\}$. For learning the model with BPR optimization criterion, we need to calculate the gradients with respect to the model parameters, which are calculated by:

$$\begin{aligned} \frac{\partial}{\partial x_u}\hat{p}_{dui} &= y_i^d \\ \frac{\partial}{\partial y_i^d}\hat{p}_{dui} &= x_u \end{aligned} \quad (13)$$

The model parameters for users $\{x_u\}_{u \in U}$ and brands $\{y_i^d\}_{i \in B_d}$ for each domain *d* are learned collectively with BPR optimization criterion. By the collectively learning, the matrix factorization model can address the first challenge to some extent. Meanwhile, by learning with BPR optimization criterion, which is a general learning framework for collaborative filtering from implicit feedbacks, the second challenge is effectively addressed.

Our Proposed Model

The matrix factorization model introduced above can address the first challenge to some extent by collective learning. However, the model ignores one important factor: the degrees of relatedness between different domains. The model just aggregates the brands in all the domains together and then factorizes the aggregated user-brand matrix, which treats the similarities between domains equally. However, the similarities between different domains are different. Thus we introduce another group of latent factors $z_d \in R^f$ to associate with each domain *d*, and the preference score $\hat{p}_{dui}$ is modeled as:

$$\hat{p}_{dui} = x_u^T y_i^d + z_d^T y_i^d \quad (14)$$

The reasons of modeling $\hat{p}_{dui}$ as above are two-fold. The first term $x_u^T y_i^d$ is the same as classical matrix factorization, which is used to model the relationship between user and brand; the second term is used to model the relationship between brand and domain, which implicitly models the relationship between different domains.

The parameters in our model are user, item and domain factors, i.e. $\Theta = \{x_u, y_i^d, z_d\}$, and the gradients of these parameters are:

$$\frac{\partial}{\partial x_u}\hat{p}_{dui} = y_i^d$$
$$\frac{\partial}{\partial y_i^d}\hat{p}_{dui} = x_u + z_d$$
$$\frac{\partial}{\partial z_d}\hat{p}_{dui} = y_i^d \tag{15}$$

The complete procedure for learning our model with BPR criterion is summarized in Table 1.

Table 1: Learning Algorithm for Our model

Procedure
For each u, draw $x_u \sim N(u, \sigma^2)$,
for each d, draw $z_d \sim N(u, \sigma^2)$,
for each $i \in B_d$, draw $y_i^d \sim N(u, \sigma^2)$.
repeat
draw (d, u, i, j) uniformly from training data T.
$\hat{p}_{duij} \leftarrow \hat{p}_{dui} - \hat{p}_{duj}$
$\delta \leftarrow c_{dui}\left(1 - \sigma\left(\hat{p}_{duij}\right)\right)$
$x_u \leftarrow x_u + \alpha\left(\delta\left(y_i^d - y_j^d\right) - \lambda x_u\right)$
$y_i^d \leftarrow y_i^d + \alpha\left(\delta(x_u + z_d) - \lambda y_i^d\right)$
$y_j^d \leftarrow y_j^d + \alpha\left(-\delta(x_u + z_d) - \lambda y_j^d\right)$
$z_d \leftarrow z_d + \alpha\left(\delta\left(y_i^d - y_j^d\right) - \lambda z_d\right)$
until convergence
end procedure

From Table 1, we can see that the learning complexity of our model is $O(f)$, which is linear to the dimension of the latent factors.

Experiments

In this section, we describe the experiments. We compare our model with three baselines: user based k-Nearest-Neighbor [Breese et al. 1998] and matrix factorization model learned with BPR [Rendle et al. 2009] for each single domain, and the matrix factorization model for multiple domains learned with BPR.

Datasets

We use two datasets for our experiments. One is a synthetic dataset from the public dataset MovieLens for movie recommendation; the other is extracted from the search log of a commercial search engine.

MovieLens. The dataset contains users' explicit ratings (from 1 to 5 stars) for movies and the genres of movies are also provided. We use four most popular genres to form the different domains. In order to better match the task of this paper, we use this dataset to mine users' preferences towards actors instead of movies. The dataset is processed in two steps. First, we extracted the actors of movies from the IMDB website and removed the movies whose actors we cannot get. Second, as our problem is tailored for implicit feedbacks, users' ratings that are greater than 2 are treated as positive feedbacks, and if a user has a positive feedback over a movie, we assume that the user has positive feedbacks over all the actors of the movie. The description of the final dataset is summarized in Table 2.

SearchLog. This dataset is extracted from the search log of a commercial search engine. We firstly predefine the brand list in four domains: digital camera, cell phone, computer, and car. Then we extracted users' search behaviors, including search queries and clicked URLs, from 10th, Nov, 2010 to 16th, Nov, 2010. If a user searched queries or clicked URLs containing the name of a brand in one of the four domains, then we think the user has positive feedbacks over the brand and has behaviors in the domain. We remove users who have behaviors in fewer than two domains. The description of the dataset is presented in Table 3.

Table 2. Description of MovieLens dataset

Domain	#Users	#Actors	#Avg. User Behaviors
Action	678	209	50.7
Comedy	725	209	53.7
Drama	828	209	54.4
Thriller	240	209	19.8

Table 3. Description of SearchLog dataset

Domain	#Users	#Brands	#Avg. User Behaviors
Digital Camera	4297	11	1.8
Cell Phone	14228	31	2.5
Computer	10743	26	1.6
Car	8889	52	3.2

Evaluation Metric

We use the leave-one-out evaluation schema, as done in [Rendle et al. 2009]. For each user u, we remove one positive feedback randomly from his behaviors in each domain, i.e. we remove one entry from P_u^d for each user u and domain d. Thus a pair of sets, namely disjoint training set D_{train}^d and test set D_{test}^d for each domain d is obtained. All the models are learned with the training data in all the domains and evaluated on all the test dataset D_{test}^d with the average AUC statistic metric:

$$\text{AUC} = \frac{1}{|D|}\sum_{d\in D}\frac{1}{|U|}\sum_{u\in U}\frac{1}{E_d(u)}\sum_{(i,j)\in E(u)}\delta\left(\hat{p}_{dui} > \hat{p}_{duj}\right)$$

where $E_d(u)$ is all the evaluation pairs in domain d for user u:

$$E_d(u) = \left\{\{\mathrm{i},\mathrm{j}\} \middle| (\mathrm{u},\mathrm{i}) \in D^d_{test} \wedge (u,j) \notin \left(D^d_{train} \cup D^d_{test}\right)\right\}$$

We repeated the experiments 10 times and the final experimental results are the average results of these experiments.

Baselines and Parameter Settings

We compare our model with the following baselines:

(1) User-KNN [Breese et al. 1998]: user-based k-nearest-neighbor method for each domain individually. The number of neighbors, k is set to 80 in our experiments.

(2) BPR-MF [Rendle et al. 2009]: Matrix factorization for each domain individually with BPR learning criterion.

(3) BPR-CMF: collective matrix factorization for all the domains with BPR learning criterion.

Our model is a collective learning method and hence we represent our proposed model with BPR-CL for short.

For all the models learned with BPR, the parameters are initialized with $\mathrm{N}(0,0.01)$; the learning rate $\alpha = 0.01$; the regularization parameter $\lambda = 0.0005$.

Results

In this subsection, we report our experimental results both on the synthetic and real world datasets.

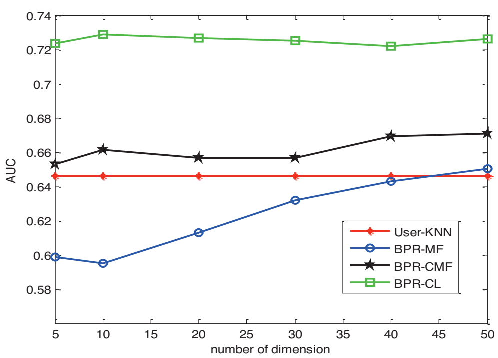

Figure 1: the performance of different methods on MovieLens dataset

Figure 1 shows the results on MovieLens dataset with different methods. We can see that our model achieves the best performance compared to the baselines. Meanwhile, we find that the collective learning methods for multiple domains (BPR-CL, BPR-CMF) all outperform the methods learned individually for each domain (User-KNN, BPR-MF). Thus we can conclude that learning multiple domains simultaneously can reduce the data sparseness in each domain.

The motivation of collective learning for multiple domains is to address the data sparseness in each domain, thus we investigate how the sparseness influences the performances of these models. The result is presented in Figure 2, where we can see that no matter how sparse the dataset becomes, our model performs consistently better than the baselines. Furthermore, the performance gain of our model increases when the data sparseness becomes serious.

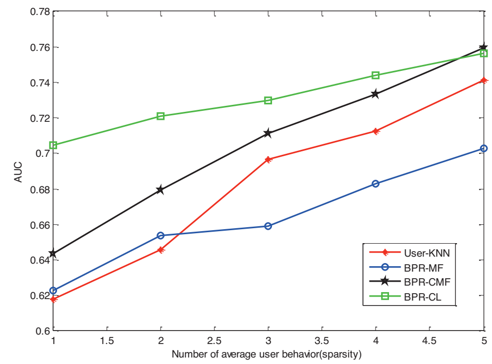

Figure 2: The influence of sparseness with different methods on MovieLens dataset

Based on the results of the synthetic dataset Movielens, we can conclude that learning multiple domains simultaneously can effectively address the data sparseness in each domain. Besides, by implicitly modeling the relationship between different domains, the proposed model performs the best among all the models and the performance gain further increase when the data sparseness becomes serious.

In order to further validate the conclusion obtained from the synthetic dataset, we repeated the experiments on the real world dataset SearchLog. The results are presented in Figure 3.

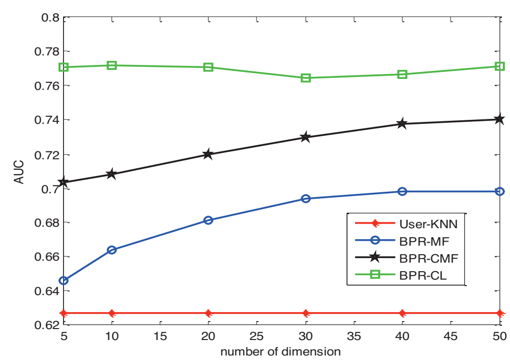

Figure 3: The performance of different methods on SearchLog dataset.

From Figure 3, we can see that on the real world dataset, our model significantly outperforms the baselines. This is because the real world dataset is quite sparse and our model can more effectively address the data sparseness problem, as proved on the MovieLens dataset. Another advantage is that our model is not sensitive to the number of latent dimension, which means we can train our model much faster and meanwhile achieve similar performance with low latent dimension.

Related Work

The related work comes from two series of work: one-class collaborative filtering and multi-relation learning.

One-class collaborative filtering. Instead of learning users' preferences from their explicit feedbacks, one-class collaborative filtering learns from their implicit feedbacks, where we can only observe the positive feedbacks. [Pan et al. 2008] proposed a weighted low-rank approximation schema for the problem, where all the missing data are treated as negative examples and a weight is assigned for each negative example for quantifying its contribution. [Hu et al. 2008] only modeled users' positive feedbacks and neglected the missing data with the classical latent factor models and a confidential value is associated with each positive example to indicate the confidence of observing it. Both of the two previous methods employed the optimization method of alternating least square [Bell et al. 2007], and the time complexity is $O(f^3)$, where f is the dimension of the latent factors. In [Rendle et al. 2009], the authors proposed to learn users' preferences from item preference pairs as training data and optimize for correctly ranking item pairs instead of scoring single items. They assume users prefer the items they viewed to those they did not and item preference pairs for each user are collected as the training data. Besides, they provide a generic optimization criterion BPR and the learning time complexity is $O(f)$. Our model also learns with the BPR optimization criterion, thus the time complexity is also $O(f)$ and is much lower than the alternating least square method.

Multi-relation learning. The multi-relation learning task aims to jointly model multiple relations [Cao et al. 2010, Li et al. 2009a, Li et al. 2009b, Singh et al. 2008, Xu et al. 2009, Zhang et al. 2010]. In [Singh et al. 2008], the authors proposed to solve the problem with the collective matrix factorization method. By sharing the parameters among the relations, the knowledge is transferred across the relations and the learning processes are mutually enhanced. Another kind of work for multi-relation learning is based on Gaussian process [Rasmussen et al. 2006]. In [Xu et al. 2009], the authors jointly modeled multiple relations with Gaussian Processes. However, they did not consider the degrees of relatedness between relations. The most related work with us is [Cao et al. 2010], which also deals with the learning of multiple sparse relations. Our paper differs from theirs in two respects. First, their problem tailors for explicit feedbacks while ours for implicit feedbacks. Second, built on Gaussian process, their model is too complex to adapt to web-scale problems; while our model learns with BPR learning criterion accompanied by the complexity linear to the dimension of latent factors, so that it can effectively be used for the users' preference mining with web-scale size.

Conclusions

In this paper, we investigated the problem of learning users' brand preferences from their implicit feedbacks. Due to users' behaviors are sparse in each domain, we propose to mine their brand preferences across multiple domains simultaneously. We proposed a latent factor model that explicitly models the relationship between users and brands and implicitly models the relationship between different domains. Experiments show that the collective learning methods outperform the methods that learn each domain individually. Especially, our proposed model significantly outperforms the baselines by modeling the similarities between domains. In the future, we plan to apply our model for mining users' other preferences.

References

Breese J.; Heckerman D.; and Kadie C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In UAI'98, 43-52.

Bell R.; and Koren Y. 2007. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In ICDM'07,43-52.

Cao B.; Liu N.; and Q. Yang. 2010. Transfer Learning for Collective Link Prediction in Multiple Heterogeneous Domains. In ICML'10,159-166.

Holland S.; Ester M.; and Kiebling W. 2003. Preference mining: a novel approach mining user preferences for personalized applications. In PKDD'03, 204-216.

Hu Y.; Koren Y.; and Volinsky C. 2008. Collaborative filtering for implicit feedback datasets. In ICDM'08,263-272.

Jung S.; Hong J. H.; and Kim T. S. 2005. A statistical model for user preference. IEEE TKDE, 2005, 834-843.

Li B.; Yang Q.; and Xue X. 2009. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In IJCAI'09, 2052-2057.

Li B.; Yang Q.; and Xue X. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In ICML'09, 617-624.

Pan R.; Zhou Y.; Cao B.; Liu N.; Lukose R.; Scholz M.; and Yang Q. 2008. One-class collaborative filtering. In ICDM'08, 502-511.

Rasmussen C.; and Williams C. 2006. Gaussian processes for machine learning. The MIT Press, 2006.

Rendle S.; Freudenthaler C.; Gantner Z.; and Schmidt-Thieme L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In UAI'09, 452-461.

Singh A. and Gordon G. J. 2008. Relational learning via collective matrix factorization. In KDD'08, 650-658.

Xu Z.; Kersting K.; and Tresp V. 2009. Multi-relational learning with Gaussian processes. In IJCAI'09, 1309-1314.

Zhang Y.; Cao B.; and Yeung D. 2010. Multi-domain collaborative filtering. In UAI'2010, 725-732.