

Multiple Attribute Aware Personalized Ranking

Weiye Guo^{1,2}, Shu Wu¹, Liang Wang¹, and Tieniu Tan¹

¹Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

²College of Engineering and Information Technology, University of Chinese Academy of Sciences, Beijing 100049, China.

weiyu.guo@ia.ac.cn, {shu.wu, wangliang, tnt}@nlpr.ia.ac.cn

Abstract. Personalized ranking is a typical task of recommender systems. It can provide a set of items for specific user and help recommender systems more correctly direct each item to its user. Recently, as the dramatically increasing social media, an entity, i.e., user and item, usually associates with multiple kinds of characterized information, e.g., explicit ratings, implicit feedbacks, and multi-type attributes (such as age, sex, occupation, or posts of user). Intuitively, comprehensively considering these information, we can obtain better personalized ranking results. However, most conventional methods only take collaborative information (explicit ratings or implicit feedbacks) or single type attributes into account. In this work, we investigate how to combine multiple attribute and collaborative information to learn the latent factors for entities and the attribute-aware mappings. As a result, we propose a novel Multiple-attribute-aware Bayesian Personalized Ranking model, Maa-BPR, for personalized ranking, which can learn reliable latent factors for entities as well as effective mappings for multiple attribute. The experimental results show that, compared with the state-of-the-art methods, Maa-BPR not only provides better ranking performance, but also is robust to new entities and the incomplete attributes.

Keywords: Personalized Ranking; Multiple Attribute; Cold Start

1 Introduction

Due to the dramatically increasing content in the Web, users are now suffering from the information overload. To cope with such heavy burden, recommender systems, which embed personalized ranking techniques, have attracted a significant attention from both academic and industrial communities. For example, Taobao and Amazon use personalized ranking techniques to improve their recommender systems in order to attract customers for more purchasing.

Among a variety of recommendation methods, the collaborative factorization is one of the most successful approaches, which learns low dimensional latent representations for both users and items. Most of work [1–3] in this field is based on explicit ratings. However, explicit ratings are generated by users actively interacting with the systems, and are hard to be obtained in practice. For instance,

users are encouraged to provide ratings for the movies on the site of MovieLens, and then the recommender system provides recommendation services based on these explicit ratings. In many scenarios, the systems can only obtain implicit interactions between users and items, e.g., whether a user viewed a web page or whether a customer purchased an item. These binary signals are called as implicit feedbacks. Recently, some factorization methods [4–6] are proposed to exploit the implicit feedbacks for personalized ranking. Typical examples, such as Bayesian Personalized Ranking (BPR) [7] and its extensions [8, 5, 1], are popular for personalized ranking by assuming that users are interested in items they had selected than the remaining items.

The above factorization methods indeed promote personalized ranking a lot. However, they are easily suffering from the cold start problem because of lacking enough collaborative information for entities. Recently, with the increasing of social media, apart from collaborative information, there is much attribute information associated with entities, such as the profiles, posts of users, and the description of items. By utilizing such information, researchers have proposed some methods [8, 9] to deal with the lack of enough collaborative information. These methods usually treat the attribute information as complete supplementary to collaborative information. However, in real-world applications, the attribute information is usually noisy, incomplete and having multiple types. How to cope with these issues is still a challenge in personalized ranking. Besides, since the implicit feedbacks are widespread and easy to be collected, how to systematically combine vast multiple types of attributes and sparse implicit feedbacks to achieve better recommendation performance is still a thorny problem.

In this paper, for enhancing the performance of personalized ranking, we propose a novel model, which can fuse multiple attribute, and jointly learn latent vectors for entities as well as effective mappings for multiple attribute. Different from the traditional attribute-aware factorization models [10, 8], which usually assume that the attribute information of entities is complete and only has single type, our proposed model is robust to the incomplete attributes and multiple attribute. Moreover, we present an advanced parameter learning algorithm, which is different from the traditional methods that segment the processes of learning latent vectors and learning attribute mappings into two independent parts. Through a unified parameter learning framework of systematically combining multiple attribute and implicit feedbacks, our parameter learning algorithm can learn more reliable latent vectors for entities as well as obtain the mappings from the multiple attribute spaces to the latent feature space.

In a nutshell, our contributions in this paper are listed as follows:

1. To systematically combine collaborative information and attribute information for personalized ranking, we propose a multiple-attribute-aware method, Maa-BPR, which can learn more reliable latent vectors for entities and obtain the attribute-aware mappings.
2. For fusing multiple attribute, we bring structured regularizers into our model, and provide a reasonable solution for jointly learning parameters.

3. For investigating the performance of our method, we conduct a series of experiments, and the results show that our method can combine multiple attribute and implicit feedbacks for improving personalized ranking.

2 Related Work

In this section, we discuss related work in two branches, i.e., collaborative methods and attribute-aware methods.

Collaborative methods [4, 7, 11] are usually based on a mass of users interactions with items, which are called as collaborative information. These methods attempt to factorize collaborative information, and map users and items into a shared latent space. Matrix factorization (MF) [12] is a classical factorization method to be used for dealing with explicit ratings. There are various extensions of MF for personalized ranking. For instance, for dealing with implicit feedbacks, implicit MF (iMF) [4] extends the basic MF by introducing adaptive confidence weights for each user-item pair. Although MF methods, e.g., iMF, can be extended to deal with implicit feedbacks, the phenomenon of data skew commonly exists in the datasets of implicit feedbacks (the number of positive samples is usually less than one percent of the total number), which causes the MF based methods to easily suffer from the over-fitting problem. Recently, Rendle et. propose a framework for personalized ranking, i.e., Bayesian Personalized Ranking (BPR) [7], which can cope with data skew in implicit feedback datasets. BPR and its extensions [5, 1, 6] make a pair-wise assumption that users are interested in items they had selected than the remaining items, which results in a pair-wise ranking object that tries to discriminate between a small set of selected items and an extremely large set of irrelevant items. Since massive training instances will be derived by the assumption, the learning of parameters is typically based on a stochastic gradient descent (SGD) with uniformly drawn pairs [6].

On the other hand, since the attribute information can indicate many characteristics of entities, attribute-aware methods [8, 2, 13] are also a kind of mainstream approaches in the field of personalized ranking. For example, in Factorization Machines (FM) [2], all kinds of attribute information are concatenated into a feature matrix, and then factors associated with attributes are learned by a process of rating regression. However, in real-world applications, the attribute information usually is multiple types, noisy and redundant. For fusing multiple attribute, [13] and [14] have been proposed, which utilize techniques of structured sparsity to handle multi-type attributes. Besides, to alleviate the cold start problem, Map-BPR [8] improve BPR framework to learn attribute-aware mappings from both collaborative information and attribute information. Note that, unlike Map-BPR that only treats the attributes as complete information with a single type, our work takes multiple, noisy and redundant attributes into account. Moreover, due to the limitation of parameter learning, the latent vectors of observed entities learned by Map-BPR only encode the collaborative information of entities, while our learning algorithm can learn the latent vectors of combining collaborative information and attribute information for entities.

3 Proposed Model

In real applications, recommender systems mainly obtain two kinds of information, i.e., collaborative information and attribute information. The information linked to a user-item pair is called as collaborative information, e.g., explicit ratings and implicit feedbacks. The information pertaining to one entity is the attribute information, e.g., profile and posts of a user, or genre, cast and description of movies, etc.

In this section, we explore these two kinds of information, and propose a new model to incorporate the collaborative information and the attribute information into a unified personalized ranking framework. Before diving into the details, we first present our model from a sketched view. Then, we will introduce how to model the implicit feedbacks and the multiple attribute respectively. Finally, we formally propose our model and give the algorithm of parameter learning.

3.1 The Sketch of Model

In traditional factorization models [4, 7, 11], every entity is represented by a latent vector, which can be learned if the entity occurs in the (collaborative) training set. However, when the models are suffering from an entity associated with few collaborative data, the latent vector of the entity could not be well learned. Furthermore, new entities without any collaborative information could be added into the real systems at any time. To employ factorization models for the entities of lacking collaborative information, a common method is to learn the attribute mappings from the attribute space to the latent feature space, and then estimate the latent vectors for these entities of lacking collaborative information by their existing attribute information.

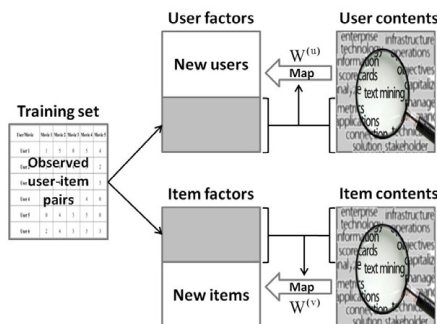


Fig. 1. The framework of combining collaborative information and attribute information into a factorization model.

Figure 1 illustrates the sketch of our factorization model. Our model is driven by three kinds of data: the collaborative training data (left), the latent factors of entities (middle) and the attributes of entities (right). The rectangles in the middle part represent the factor matrices, where the entities of lacking collaborative data have no latent factors. These unknown latent factors will be estimated using

the corresponding attribute mappings. Thus, taking the attribute information into account, the training of our model consists of the following alternant learning steps: learning the latent vectors of entities from the collaborative information and the attribute information; then, learning the attribute-aware mappings from the latent vectors of observed entities and their attributes.

3.2 Learning Latent Factors From Implicit Feedbacks

BPR [7] is a popular personalized ranking framework for dealing with the implicit feedbacks. In the BPR framework, if the user u_m has selected the item v_i but not selected the item v_j , then the work of BPR assumes that, u_m prefers v_i over v_j , and defines the pairwise preference of u_m as

$$p(i \succ_m j) := \Phi(x_{mij}), \quad (1)$$

where $\Phi(x) = 1/(1 + \exp(-x))$ and $x_{mij} := r(u_m, v_i) - r(u_m, v_j)$. $r(\cdot, \cdot)$ can be any kind of scoring functions which indicate the relevance between two entities.

Because this work mainly focuses on studying the implicit feedbacks, we simply follow the pair-wise assumption of BPR [7] to create training data $D_S := \{(m, i, j) | v_i \in I_{u_m}^+ \wedge v_j \in I \setminus I_{u_m}^+\}$ from collaborative training data, where $I_{u_m}^+$ denotes the set of the observed items which are linked to the user u_m , and the triple $t = (m, i, j)$ in D_S represents the user u_m is relevant to the item v_i but irrelevant to the item v_j . For simplicity, we call v_i as a positive item of u_m , while v_j is a negative item of u_m . It should be noted that the training triples can be easily created from any datasets with explicit ratings. For example, if the user u_m has given a higher score to the item v_i than the item v_j , then we can treat v_i as the positive item of u_m and v_j as the negative one.

After obtaining the training triples, the goal of BPR is to maximize the likelihood of all pair-wise preference:

$$\arg \max_{\Theta} \prod_{(m,i,j) \in D_S} p(i \succ_m j), \quad (2)$$

which is equivalent to minimize the negative log likelihood:

$$L_{feedback} = - \sum_{(m,i,j) \in D_S} \ln \Phi(x_{mij}) + \lambda \|\Theta\|^2, \quad (3)$$

where Θ is the latent feature vectors of entities and λ is a hyper-parameter.

3.3 Learning Attribute-Aware Mappings

In the above sections, we have illustrated how to learn the latent factors of entities only considering implicit feedbacks. However, in the real recommender systems, entities are not always having enough collaborative information, e.g., new entities will be added into the systems at any time. Therefore, apart from the collaborative information, we also need to take the attribute information of entities into account for learning latent vectors of entities. For the sake of simply bridging the attribute space and the latent space, as shown in Figure 2, we concatenate the multiple attribute of an entity into an attribute vector, and

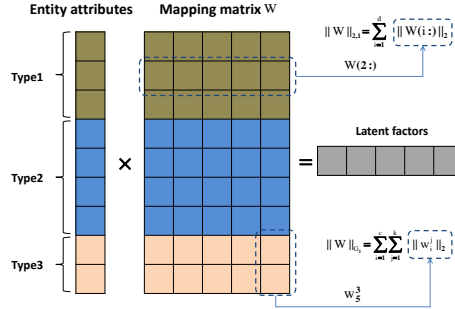


Fig. 2. The schematic diagram of mapping from the attribute space to the latent feature space, where we assume an entity contains three different types of attributes. We concatenate all attributes into an attribute vector, and map the attribute vector to be the latent factors by the attribute-aware mapping matrix. Besides, for dealing with the multiple attribute which is noisy, redundant and consists of multiple types, G_1 -norm and $l_{2,1}$ -norm are used for regularization on the mapping matrix.

then leverage a linear mapping to map the attribute vector to be a latent vector. Thus, our task is in turn to learn the linear mappings for attribute information.

Here, we formally present an unsupervised solution for learning the linear mappings. In this work, we consider the classical scenario of recommendation, which has two types of entities, i.e., users (e.g., customers) and items (e.g., movies, books and songs). We use u and v to denote a user and an item, respectively. For simplicity, we use e to denote an abstract entity, which can be a user or an item. $y_i^{(e)}$ denotes the latent vector of the entity e_i and the matrix $Y^{(e)} = [y_1^{(e)}, y_2^{(e)}, y_3^{(e)} \dots]$ is the latent vectors of all the users/items. For example, $y_m^{(u)}$ denotes the latent vector of user u_m and $Y^{(u)}$ denotes the latent matrix of user. Besides, we use the matrix $A^{(e)} = [a_1^{(e)}, a_2^{(e)}, a_3^{(e)}, \dots]$ to denote the multiple attribute of entities. $a_i^{(e)} \in \mathbb{R}^d$ is an attribute vector concatenating k types of attributes, where d is the total number of attributes of an entity, each type j has d_j attributes, and $d = \sum_{j=1}^k d_j$. Then, we can present the objective function to learn the attribute-aware mappings from the attributes of k types:

$$L_{attribute} = \|A^{(e)}W^{(e)} - Y^{(e)}\|_F^2, \quad (4)$$

where $W^{(e)} = [w_1^1, \dots, w_c^1; \dots, \dots, \dots; w_1^k, \dots, w_c^k] \in \mathbb{R}^{d \times c}$ denotes a mapping matrix, and c is the dimension of latent vectors. w_p^q indicates the weights of all attributes in the q -th type with respect to the p -th latent factor.

However, the optimization problem expressed by Eq. (4) has infinite solutions because of lacking supervised information. Fortunately, using the BPR framework, we can obtain an approximate latent matrix $\widetilde{Y}^{(e)}$ for $Y^{(e)}$ from the collaborative information. As a result, we can treat $\widetilde{Y}^{(e)}$ as pseudo labels, and substitute it into Eq. (4). With the pseudo labels, we can solve the optimization problem, and jointly learn the mapping matrix $W^{(e)}$ from both the collaborative information and the attribute information.

However, the attributes of entities usually are multiple types, noisy and redundant. Taking these issues into considerations, we still need to design a proper scheme to deal with the interrelations among multiple attribute, and select the informative attributes from a mass of attributes.

In multiple attribute fusion, different types of attributes can be more or less discriminative for different factors. For instance, the description of a movie is usually more associated with the genre of the movie, while the cast of this movie indicates the potential popularity of the movie. To capture the interrelations between multiple attribute and latent factors, we attempt to conduct structured sparsity on mapping matrices. Thus, the group l_1 -norm (G_1 -norm) [15] is used for regularization, which is defined as $\|W\|_{G_1} = \sum_{i=1}^c \sum_{j=1}^k \|w_i^j\|_2$. As illustrated in Figure 2, the G_1 -norm uses l_2 -norm within each type of attributes and l_1 -norm between types. In this way, the sparsity between different types is enforced [15], i.e., if attributes of one type are not discriminative for the latent factors of a certain group, the elements corresponding to those latent factors in $W^{(e)}$ will be assigned with zeros (in practical case, they are usually very small values), otherwise, their weights should be large.

Besides, in certain cases, even if most attributes of one type are not discriminative for the latent factors of any groups, there are still a small number of attributes in this type to be highly discriminative. Such important attributes should be shared by all latent factors. Thus, the $l_{2,1}$ -norm [16] is also used for regularization, which is defined as $\|W\|_{2,1} = \sum_{i=1}^d \|W(i :)\|_2$, where $W(i :)$ is the i -th row of matrix W . Thus, the linear mappings can be learned by

$$L_{attribute} = \sum_{e \in \{u,v\}} \lambda^{(e)} (\|W^{(e)} A^{(e)} - \widetilde{Y}^{(e)}\|_F^2 + \|W^{(e)}\|_{G_1} + \|W^{(e)}\|_{2,1}), \quad (5)$$

where $\lambda^{(e)}$ is a hyper-parameter for tuning the weight of users or items.

3.4 The Unified Model and Parameter Learning

To systematically incorporate collaborative information and attribute information into a solution, our model for learning latent vectors and attribute-aware mappings can be expressed as

$$\begin{aligned} & \arg \min_{\Theta, W} L_{feedback} + L_{attribute} = \\ & - \sum_{(m,i,j) \in D_s} \ln \Phi \left((y_i^{(v)} - y_j^{(v)}) y_m^{(u)} \right) + \lambda \|\theta\|^2 \\ & + \sum_{e \in \{u,v\}} \lambda^{(e)} \left(\|W^{(e)}\|_{G_1} + \|W^{(e)}\|_{2,1} + \|A^{(e)} W^{(e)} - Y^{(e)}\|_F^2 \right). \end{aligned} \quad (6)$$

To learn the parameters $Y^{(u)}$, $Y^{(v)}$, $W^{(u)}$ and $W^{(v)}$ in Eq. (6), we design an alternative optimization algorithm, which uses SGD with uniformly drawn training triples to learn the latent vectors and implements matrix decomposition to learn the mapping matrices.

In each iteration, when we are updating the latent factor matrix $Y^{(e)}$, we set the mapping matrix $W^{(e)}$ to be a constant and $L_{attribute}$, the entire optimization

objective of attribute information, to be a regularizer. Thus, the gradient of an arbitrary latent parameter θ is

$$\begin{aligned} \frac{\partial L}{\partial \theta} = & \sum_{(m,i,j) \in D_s} \left(1 - \Phi\left((y_i^{(v)} - y_j^{(v)})y_m^{(u)}\right)\right) \frac{\partial((y_i^{(v)} - y_j^{(v)})y_m^{(u)})}{\partial \theta} \\ & + \frac{\partial \sum_{e \in \{u,v\}} \lambda^{(e)} \left(\|A^{(e)}W^{(e)} - Y^{(e)}\|_F^2\right)}{\partial \theta} + \lambda\theta. \end{aligned} \quad (7)$$

The updating rule for parameter θ is $\theta = \theta + \eta \frac{\partial L}{\partial \theta}$, where η is the learning rate.

On the other hand, given a latent factor matrix $Y^{(e)}$, we view $Y^{(e)}$ as pseudo labels and treat $L_{feedback}$ as a constant. Thus, the optimization objective of Eq. (6) is equal to $L_{attribute} = 0$ and the updating rule for $W^{(e)}$ can be derived as

$$w_i = (A^{(e)}(A^{(e)})^T - \lambda D^i + \gamma \tilde{D})^{-1} A^{(e)} Y^{(e)}(i :), \quad (8)$$

where $Y^{(e)}(i :)$ is the i -th row of matrix $Y^{(e)}$. D^i is a block diagonal matrix with the j -th diagonal block as $\frac{1}{2\|w_i^j\|_2} I_j$. I_j is an identity matrix with size of d_j . \tilde{D} is a diagonal matrix with the j -th diagonal element as $\tilde{D}(j, j) = \frac{1}{2\|W^{(e)}(j :)\|_2}$. Note that both $D^i (1 \leq i \leq c)$ and \tilde{D} are dependent on $W^{(e)}$. Thus, they are also unknown variables but can be approximatively calculated by the value of $W^{(e)}$ in last iteration.

In a nutshell, the iterative algorithm for learning parameters of Maa-BPR is summarized in Algorithm 1. The algorithm mainly repeats two learning steps until the parameters reach convergence, i.e., it first learns the latent factors of entities from implicit feedbacks, by SGD with uniformly drawn training triples, then it learns the attribute mapping matrices by given latent factors.

4 Experiment

In this section, we perform experiments to validate our proposed model by comparing with other approaches on real world datasets. In the following experiments, we first investigate the comprehensive prediction quality by evaluating Area Under the ROC Curve (AUC) and Mean Average Precision (MAP). Then, we study the performance of our method on Top-N recommendation. Finally, we study the cold start problem by simulating the recommendation of new items.

4.1 Datasets

For evaluating our method, we use two real world datasets, i.e., DBLP¹ and MovieLens², and carry out training and testing on randomly split training (80%) and testing (20%) data.

DBLP contains 2,084,055 papers with 2,244,018 citations. Each paper may be associated with abstract, authors, published year, and title. We preprocess the DBLP data to be an experimental data set in a similar way as [6]. More

¹ <http://arnetminer.org/citation>

² <http://grouplens.org/datasets/movielens>

Algorithm 1 Learn parameters for Maa-BPR

Input:
 The triple set D_s ;
 The content feature of entities $A := \{A^{(u)}, A^{(v)}\}$;

Output:
 $\Theta := \{Y^{(u)}, Y^{(v)}\}$, $W := \{W^{(u)}, W^{(v)}\}$;

- 1: Initialize Θ and W ;
- 2: **repeat**
- 3: uniformly draw a triple (m, i, j) from D_s ;
- 4: **for** each latent vector $\theta \in \text{triple}(m, i, j)$ **do**
- 5: $\theta \leftarrow \theta + \eta \frac{\partial L}{\partial \theta}$;
- 6: **end for**
- 7: **for** each $W^{(e)} \in W$ **do**
- 8: Calculate the block diagonal matrix \tilde{D} ;
- 9: Calculate the block diagonal matrices $D^i (1 \leq i \leq c)$;
- 10: **for** each column $w_i (1 \leq i \leq c) \in W^{(e)}$ **do**
- 11: $w_i \leftarrow (A^{(e)}(A^{(e)})^T - \lambda D^i + \gamma \tilde{D})^{-1} A^{(e)} Y^{(e)}(i :)$;
- 12: **end for**
- 13: **end for**
- 14: **until** convergence

specifically, we sample 1,000 authors who have published no more than 5 papers and cited 5-100 papers from the DBLP data. Thus, we obtain 1,000 authors, 16,313 papers and 23,506 author-paper pairs. Each author-paper pair denotes a relation of the author cited the paper. In the experiments, we treat texts in published papers of an author as the content information of this author, and paper text as content information of the paper. We use the term-frequency over texts as features of content information. Our task is to predict the personalized ranking of citing papers for each author.

MovieLens includes 100,000 ratings by 943 users on 1682 movies. Each user has rated at least 20 movies. In our experiments, the age, gender and occupation of a user are used as the multiple attribute of the user, and the genre of movie and key words in title are viewed as the multiple attribute of an item. Using the same processing method in [17], we do not use the rating values but just binary rating events by assuming that users tend to rate movies they have watched. For a specific user, our task is to predict the potential ranking list of movies.

In addition, comparing these two experimental datasets, we can observe that: 1) the implicit rating matrix of DBLP data set is sparser than that of MovieLens; 2) each entity in MovieLens has its corresponding attributes, but entities in DBLP always lack corresponding texts. Thus, in DBLP, many authors and papers have incomplete attribute information.

4.2 Compared Methods

Table 1 shows the characteristics of compared methods. Our method, i.e., Maa-BPR is first compared with two basic methods, i.e., BPR-MF [7] and iMF

[6], which only consider implicit feedbacks. Then, we investigate two advanced attribute-aware methods, i.e., Map-BPR [8] which extends BPR with attribute mappings and FM [2] which is an attribute-aware framework. In our experiments, since the datasets we used existing data skew, i.e., the number of positive feedbacks is far less than the number of negative feedbacks, we train FM and MF with the training data sets which contain randomly drawn negative feedbacks, i.e., the proportions of negative feedbacks and positive feedbacks are 50 : 1 on DBLP and 100 : 1 on MovieLens.

Table 1. The characteristics of compared methods

| Method | Attribute | Feedback | Cold start |
|----------------|------------|-------------------|------------|
| BPR-MF | no | implicit | no |
| iMF | no | implicit | no |
| FM | yes | implicit/explicit | yes |
| Map-BPR | yes | implicit | yes |
| Maa-BPR | yes | implicit | yes |

4.3 Results and Discussion

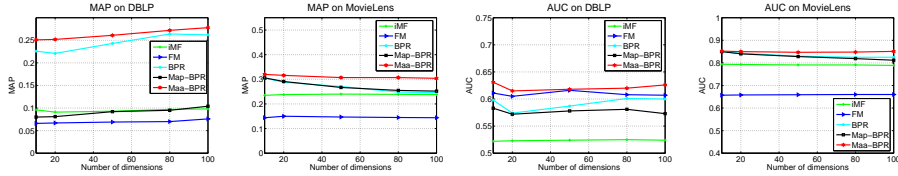


Fig. 3. The comprehensive prediction quality, i.e., Mean Average Precision (MAP) and Area under the ROC curve (AUC), on DBLP and MovieLens.

Figure 3 presents the ranking performance of methods evaluated by MAP and AUC. Since the attribute information is noisy and consists of multiple types, the traditional attribute-aware methods, e.g., FM, usually could not get reliable results. On the other hand, due to the lack of enough collaborative information, the methods only considering implicit feedbacks, e.g., iMF and BPR-MF, also could not get ideal results. Owing much to combining multiple attribute and implicit feedbacks, Maa-BPR consistently outperforms other methods. Furthermore, we observe that, although both Map-BPR and Maa-BPR combine attribute information and collaborative information to learn the attribute mappings, the performance of Maa-BPR on different datasets is more stable than these of Map-BPR. This result demonstrates that, our method is more robust than Map-BPR in terms of incomplete attribute information. This is because that our method fuses multiple attribute, and jointly learns latent vectors from implicit feedbacks and multiple attribute.

Figure 4 shows the precision of different methods with varying numbers of recommendations. Our method achieves the best performance in most cases, especially in recommending a small set of items. Since in real-world scenarios, users only care about several items which are listed on the top places, this experiment of Top-N recommendation in turn shows that our method can well fit recommender systems. In addition, with the number of recommendations increasing,

the performance of attribute-aware methods decline more quickly than the methods only taken collaborative information into account. This phenomenon may indicate that the attribute information usually has more noise than collaborative information. Thus, it is valued for designing proper schemes to alleviate the noise of attributes, when modeling the attributes of entities.

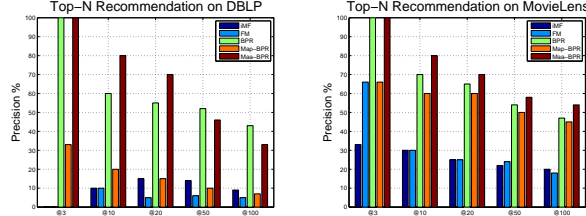


Fig. 4. The performance of Top-N recommendation on DBLP and MovieLens.

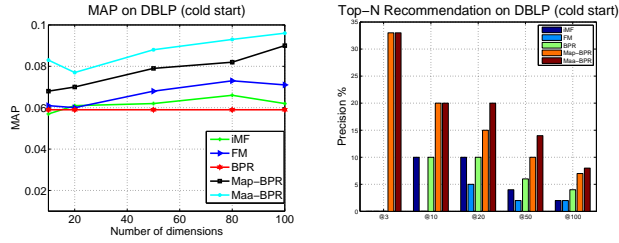


Fig. 5. The performance of new items recommendation on the DBLP dataset. We only evaluate the user-item pairs when the items are not occurring in the training set.

For investigating the cold start problem, we assess the performance of new items recommendation in Figure 5. In the experiments, We only evaluate the user-item pairs which their items are not occurring in the training set. Due to the lack of collaborative information, the traditional methods without considering attribute information can only provide random recommendation results, while most attribute-aware methods have better performance regardless of comprehensive prediction and Top-N recommendation. Moreover, Maa-BPR obtains the best results among attribute-aware methods. Besides, we can observe that FM has poor performance on the experiment of Top-N recommendation, especially in recommending a small set of items. This result may be because that FM only takes attributes into account, and Top-N recommendation is easily influenced by the noise in attributes.

5 Conclusions

For promoting personalized ranking by fusing multiple attribute, this paper has proposed a novel personalized ranking model, Maa-BPR, which combines multiple attribute and interactions between entities to learn latent vectors for entities and attribute-aware mappings for multiple attribute. Comprehensive experiments have shown that our model achieves better predictive performance and is

robust to both the cold start problem and incomplete attributes. However, the strategy of randomly drawn training triples would cause a slow convergence. In the future, to speed up the parameter learning process and further promote the performance of our model, we plan to improve the sampling strategy by taking attribute information of items and social information of users into account.

6 Acknowledgments

This work is jointly supported by National Basic Research Program of China (2012CB316300), and National Natural Science Foundation of China (61175003, 61202328, 61420106015, U1435221, 61403390).

References

1. Shuang Qiu, Jian Cheng, Ting Yuan, and Hanqing Leng. Item group based pairwise preference learning for personalized ranking. In *SIGIR*, pages 1219–1222, 2014.
2. Steffen Rendle. Factorization machines with libFM. *TIST*, 3(3):57:1–57:22, 2012.
3. Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys*, pages 269–272, 2010.
4. Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.
5. Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, pages 2691–2697, 2013.
6. Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, pages 273–282, 2014.
7. Steffen Rendle, Christoph Freudenthaler, and Lars Gantner. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
8. Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*, pages 176–185, 2010.
9. István Pilászy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *RecSys*, pages 93–100, 2009.
10. Tianqi Chen, Zhao Zheng, Qiuxia Lu, Weinan Zhang, and Yong Yu. Feature-based matrix factorization. *arXiv preprint arXiv:1109.2271*, 2011.
11. Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. Recommendation in heterogeneous information networks with implicit user feedback. In *RecSys*, pages 347–350, 2013.
12. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
13. Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. Unsupervised feature selection for multi-view data in social media. In *SDM*, pages 270–278, 2013.
14. Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *ICML*, pages 352–360, 2013.
15. Hua Wang, Feiping Nie, Heng Huang, and Chris Ding. Heterogeneous visual features fusion via sparse multimodal machine. In *CVPR*, pages 3097–3102, 2013.
16. Guillaume Obozinski and Michael I Taskar. Joint covariate selection and joint subspace selection for multiple classification problems. *ASC*, 20(2):231–252, 2010.
17. Asela Gunawardana and Christopher Meek. A unified approach to building hybrid recommender systems. In *RecSys*, pages 117–124, 2009.