Neurocomputing 214 (2016) 191–200

Contents lists available at ScienceDirect

Neurocomputing

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

Personalized ranking with pairwise Factorization Machines

Weiyu Guo [a,b], Shu Wu [a,*], Liang Wang [a], Tieniu Tan [a]

[a] Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, PR China

ARTICLE INFO

Article history:
Received 30 December 2015
Received in revised form
25 March 2016
Accepted 29 May 2016
Available online 9 June 2016

Keywords:
Personalized ranking
Adaptive sampling
Pairwise learning

ABSTRACT

Pairwise learning is a vital technique for personalized ranking with implicit feedback. Given the assumption that each user is more interested in items which have been previously selected by the user than the remaining ones, pairwise learning algorithms can well learn users' preference, from not only the observed user feedbacks but also the underlying interactions between users and items. However, a mass of training instances are randomly derived according to such assumption, which makes the learning procedure often converge slowly and even result in poor predictive models. In addition, the cold start problem often perplexes pairwise learning methods, since most of traditional methods in personalized ranking only take explicit ratings or implicit feedbacks into consideration. For dealing with the above issues, this work proposes a novel personalized ranking model which incorporates implicit feedback with content information by making use of Factorization Machines. For efficiently estimating the parameters of the proposed model, we develop an adaptive sampler to draw informative training instances based on content information of users and items. The experimental results show that our adaptive item sampler indeed can speed up our model, and our model outperforms advanced methods in personalized ranking.

1. Introduction

Due to information overload on the Web, both users and e-commerce sellers, such as Taobao and Amazon, expect to exactly direct appropriate items to specific users. Consequently, recommender systems with personalized ranking have become an important part in modern applications. For example, Taobao and Amazon embed personalized ranking techniques into their recommender systems, which navigate potential customers to products according to the specific preference of customers. Most of the studies [1–3] in personalized ranking are based on explicit ratings. However, as users are required to actively interact with systems, explicit ratings are hard to be obtained in practical applications. For instance, MovieLens invites a mass of users to rate movies on its website, and then personalized recommendation is provided based on these explicit ratings. While implicit feedbacks, such as whether a user has browsed a web page, or whether a customer has purchased a product, pervade on the Web and are easier to be collected. We thereby extend our research to consider implicit feedbacks.

A characteristic of implicit feedbacks is that it is one-class, i.e., only positive instances are observed. Traditional Collaborative Filtering (CF) approaches [4,5] often suffer from over-fitting problem when they are dealing with this kind of data, because the number of observed positive instances is far less than that of negative instances. Pairwise learning algorithms, such as Bayesian Personalized Ranking (BPR) [6] and its extensions [3,7,8], are tailored to personalized ranking with implicit feedbacks. They usually assume that users are more interested in items that they have selected than the remaining items, and randomly draw item pairs with corresponding users to make up training instances. However, in practice, most of the users have only seen a handful of items, and provided feedbacks on some of viewed items. Consequently, when the number of items is very large, the random sampler will derive massive meaningless and ineffective training instances which cannot provide much helpful information to tune the personalized ranking lists. For example, in common sense, we cannot say that a user dislikes a smart phone because he likes a toothbrush, and this pair, i.e., toothbrush and smart phone, is meaningless in real-world scenarios. Moreover, users cannot view too many items at the same time, but the random sampler often draws item pairs which items are not viewed by corresponding users. For instance, when a user searches smart phones on an e-commerce website, the smart phones of popular brands may be viewed simultaneously by the user. While the smart phones of non-mainstream brands, usually are ranked at tail places of the recalled list, and the user tends to miss such non-mainstream

* Corresponding author.
E-mail addresses: weiyu.guo@ia.ac.cn (W. Guo), shu.wu@nlpr.ia.ac.cn (S. Wu), wangliang@nlpr.ia.ac.cn (L. Wang), tnt@nlpr.ia.ac.cn (T. Tan).

http://dx.doi.org/10.1016/j.neucom.2016.05.074
0925-2312/

brands. Due to the lack of user feedbacks, the predictive models are hard to accurately assess the degree of the user interest in non-mainstream smart phones, and the corresponding training instance has a low contribution to the learning of parameters. The slow convergence in the training procedure seems to be inevitable, on account of wasting a lot of time on massive meaningless and ineffective training instances.

To the best of our knowledge, there are two directions to design efficient samplers for speeding up the procedure of pairwise learning, i.e., directly utilizing characteristics of distribution appearing in datasets, e.g., long-tailed distribution, to enlighten the samplers [9], or leveraging adaptive strategies [9,10] to adaptively guide the samplers. Since different datasets may follow different distributions, the methods in the first direction are hard to be scaled to general applications. Therefore, we turn to speed up the procedure of pairwise learning by the second direction. We design an adaptive item sampler which holds a self-evident rule, i.e., "apples to apples", for drawing informative training instances. For example, suppose that a user has bought an iPhone 6, our sampler tends to draw other kinds of smart phones which are able to be compared with iPhone 6 in terms of product properties, rather than draw other things, e.g., toothbrushes. Moreover, under the category of smart phones, we further select those products which probably have been viewed by the user to make up the item pairs for the user.

On the other hand, new users and items may be added into recommender systems consistently. The conventional pairwise learning algorithms for personalized ranking, which only take account of collaborative information, e.g., implicit feedbacks or explicit ratings, are only able to deal with entities observed in the dataset. For new users or items which do not have any collaborative information, such methods are not capable of making meaningful personalized ranking. In real-world recommender systems, such cold start problems are often addressed by switching to content-based approaches.

In this work, for obtaining robust personalized ranking results, we associate content information of entities with implicit feedbacks to develop a Pairwise Ranking Factorization Machines (PRFM) which alleviates the cold start problem and enhances the performance of personalized ranking by incorporating BPR learning [6] with Factorization Machines [2]. Furthermore, for a comprehensive solution, we embed an adaptive and efficient sampler into the learning procedure of PRFM by making use of the parameters of PRFM and the content information of users and items.

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

- We associate content information of entities with implicit feedbacks to develop a pairwise learning framework, PRFM, which not only can alleviate the cold start problem, but also can improve personalized ranking by combining content information and implicit feedbacks.
- To speed up the learning of PRFM, we take account of the content information, and develop an adaptive sampling strategy to draw informative training data. Our adaptive sampler provides a better balance between efficiency and performance than previous strategies.
- We conduct a series of experiments to validate the proposed methods. The results show that PRFM can improve the performance of personalized ranking.

2. Related work

Factorization methods are popular in personalized recommender systems. They are utilized to deal with various information collected by recommender systems, such as user feedbacks [6,11], attributes of item [2,7], user profiles [12] and social information [13]. According to the studied data types, these methods can be roughly arranged into two branches, i.e., collaborative methods and content-based methods.

2.1. Collaborative methods

Collaborative Filtering (CF) has been one of the most successful approaches in recommender systems [4,5,14]. It collects users' historical data and generates predictions for such users based on similarity measurements of users and items. Matrix factorization (MF) techniques, such as SVD [15] and SVD ++ [16], are very popular for collaborative filtering based on explicit ratings. They factorize the rating matrix to be two low rank matrices, which can indicate users' preference and properties of items respectively.

Although some extensions [17–19] of MF can deal with implicit feedbacks, they easily suffer from the over-fitting problem because of commonly existed data skewness in implicit feedback datasets (the number of positive feedbacks is usually less than one percent of the total number). For alleviating the data skewness and providing personalized ranking lists based on implicit feedbacks, Bayesian Personalized Ranking (BPR) [6] and its extensions [3,8,9] are proposed, which make a pairwise assumption that users are more interested in items that they have previously selected than the remaining items.

Since the pairwise assumption of BPR often derives a large-scale training set, the randomly sampling strategy [6] is popular to draw training instances. However, the random sampler often leads to slow convergence. To speed up the BPR learning, Rendle et al. utilize the properties of long-tail effect to develop non-uniformly item samplers [9]. Zhong et al. [10] draw informative training pairs according to the preference of a given user on two unselected items. However, since the number of items in real-world datasets is very large, previous works [9,10] are perplexed by the dilemma in terms of how to balance the efficiency of sampler and the predictive quality of model. To alleviate this dilemma, Guo et al. [20] designed an adaptive sampler which uses feature learning methods to learn low dimensional representations for items, and utilizes such representations to initialize the ranking scores of items under categories. However, this strategy may depend on the quality of initialization too much. If the feature learning method cannot obtain high-quality representations for items, then this kind of adaptive sampler may suffer from a bad performance.

2.2. Content-based methods

Content-based methods [2,21,22] utilize contents of entities, such as attributes of items, texts of users and reviews of items, to learn factors of contents and predict specific users' preference.

Factorization Machines (FM) [2] are a generic predictor, since they can mimic most factorization models with feature engineering [2]. In FM, all kinds of contents are concatenated to be a design matrix, and then factors associated with contents are learned by a process of regression or classification. Recently, for providing an easy access to different solvers for regression, classification and ranking tasks, fastFM [23] embeds the various losses into the FM framework. For instance, to deal with the ranking task, fastFM uses a random sampler to draw item pairs for the corresponding users, and learns the parameters of FM by making use of the pairwise loss of BPR. However, fastFM with pairwise loss of BPR cannot well learn some parameters of FM which are relevant to user features, and this limitation hampers it to improve the quality of personalized ranking.

On the other hand, to alleviate the cold start problem, Map-BPR [7] and Maa-BPR [24] extend the BPR framework to learn content-

aware mappings from the content space to the latent space. However, Map-BPR segments the procedures of matrix factorization and the learning of content-aware mappings to two independent phases. This limitation causes that the low rank matrices produced by Map-BPR ignore content information of user and item. Maa-BPR improves Map-BPR, and incorporates multiple attributes with implicit feedbacks to learn latent vectors for entities and attribute-aware mappings for multiple attributes.

3. Background

In this section, we introduce Bayesian Personalized Ranking (BPR) [6] and Factorization Machines (FM) [2], inspired by which we construct our model.

3.1. Bayesian Personalized Ranking

Bayesian Personalized Ranking (BPR) [6] makes a pairwise assumption that users are more interested in items that they have selected than the remaining items to learn low rank matrices for users and items respectively. Specifically, if user u_m has selected item v_p but not selected item v_q, then BPR assumes that u_m prefers v_p over v_q, and defines this pairwise preference of u_m as

$$P(p \succ_m q) := \delta(x_{mpq}), \tag{1}$$

where $\delta(x) = 1/(1+\exp(-x))$ and $x_{mpq} := f(u_m, v_p) - f(u_m, v_q)$. $f(\cdot,\cdot)$ is a scoring function which indicates the relevance between a user and an item.

Based on the pairwise preference assumption, the set of all pairwise preference $D_S := \{(m, p, q) | v_p \in I_{u_m}^+ \wedge v_q \in I \backslash I_{u_m}^+\}$ can be created from the corresponding implicit feedback dataset, where $I_{u_m}^+$ denotes the set of items which have been selected by user u_m, and a triple $t = (m, p, q)$ represents that user u_m is relevant to item v_p but irrelevant to item v_q. For simplicity, we can call v_p as a positive item of u_m, while v_q is a negative one. And (u_m, v_p) is a positive feedback, (u_m, v_q) is a negative one.

Given a set of pairwise preference D_S, the optimization goal of BPR learning is to maximize the BPR-OPT criterion:

$$BPR-OPT := \ln \prod_{(m,p,q)\in D_S} P(p \succ_m q), \tag{2}$$

which is equivalent to minimize the negative log likelihood:

$$L = -\sum_{(m,p,q)\in D_S} \ln \delta\left(x_{mpq}\right) + \frac{\lambda}{2} \| \Theta \|^2, \tag{3}$$

where Θ denotes the set of parameters and λ is a hyper-parameter. Since the size of D_S is very large, the learning algorithms of BPR are generally based on stochastic gradient descent (SGD) with randomly drawn training data from D_S.

3.2. Factorization Machines

Factorization Machines (FM) [2] is a generic method, which can mimic most of factorization models just using feature engineering [2].

In FM, all kinds of contents are concatenated into a design matrix $\mathbf{X} = [x_1, x_2, \ldots, x_n]$, where x_i is the feature vector of the ith sample, and n is the number of training samples. Then, the FM model of order 2 is defined as

$$\hat{y} := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \sum_{k=1}^{K} v_{ik} v_{jk}, \tag{4}$$

where $\hat{y}$ is the predictive value. w_0 is the global bias, and w_i is the strength of the ith feature. v_{ik} is the kth interaction factor for the ith feature. K is the number of interaction factors for each feature.

4. Proposed model

In this section, we construct a personalized ranking model, i.e., Pairwise Ranking FM (PRFM), which incorporates implicit feedbacks with contents of users and items for personalized ranking.

4.1. Problem statement

Suppose we have M users and N items in an implicit feedback dataset. Let $\mathbf{X}^U \in \mathbb{R}^{M\times f_u}$ and $\mathbf{X}^V \in \mathbb{R}^{N\times f_v}$ be the design matrices of M users and N items respectively. By making use of implicit feedbacks and content information, our goal is to build a ranking model. For each user, our model generates an optimal ranking list from the candidate set, s.t. the items in which a specific user is interested appear at the top of the list.

4.2. Model construction

For modeling implicit feedbacks and content information in personalized ranking problem, we construct the Pairwise Ranking FM (PRFM). Specifically, as shown in Fig. 1, for a user-item feedback (positive or negative), we concatenate the user features with the item features to be a feature vector of this user-item feedback. Thus, we can simply design a pairwise loss for the ranking task under the FM framework as:

$$L = \sum_{(m,p,q)\in D_s} -\ln \delta(\hat{y}_{mp}(x_{mp}|\Theta) - \hat{y}_{mq}(x_{mq}|\Theta)) + \frac{\lambda}{2} \|\Theta\|^2, \tag{5}$$

where $\hat{y}_{mp}$ and $\hat{y}_{mq}$ are preference scores which reflect the degree of user m interested in corresponding items, i.e., p and q. x_{mp} and x_{mq} are the feature vectors of user-item feedbacks (m, p) and (m, q) respectively. $\Theta := \{w_0, \mathbf{w}^U, \mathbf{w}^V, \mathbf{V}^U, \mathbf{V}^V\}$ is the parameters of FM. w_0 is the global bias of features. $\mathbf{w}^U \in \mathbb{R}^{f_u}$ and $\mathbf{w}^V \in \mathbb{R}^{f_v}$ are the strengths of user features and item features, respectively. $\mathbf{V}^U \in \mathbb{R}^{f_u\times k}$ and $\mathbf{V}^V \in \mathbb{R}^{f_v\times k}$ are interaction factors of user features and item features respectively. f_u (or f_v) is the number of user (or item) features.

Similar to BPR, we can use the Stochastic Gradient Descent (SGD) with randomly drawn training triples to solve the optimization problem in Eq. (5). However, the gradients of the global bias w_0 and the strengths of user features $\mathbf{w}^U$ are equal to 0. Thus, the global bias w_0 and the strengths of user features $\mathbf{w}^U$ cannot be well learned in the training procedure. Specifically, for an arbitrary strength of feature $w_i \in \Theta$, its gradient is

$$\frac{\partial L}{\partial w_i} = (1 - \delta(\hat{y}_{mp}(x_{mp}|\Theta) - \hat{y}_{mq}(x_{mq}|\Theta))(x_{mpi} - x_{mqi}) + \lambda w_i, \tag{6}$$

where x_{mpi} (or x_{mqi}) is the ith feature of a corresponding user–item feedback. As it can be seen in Eq. (6), when w_i refers to a user feature dimension, x_{mpi} is equal to x_{mpj}. Thus, the gradient of w_i is 0 and any information cannot be learned from the feature x_{mpi} (or x_{mqi}) for w_i.

To overcome the gradient vanishing on the strengths of user features and the global bias, we propose the Pairwise Ranking FM (PRFM) model, which improves Eq. (5) by constructing two FM regressors, and defines a new pairwise loss as:

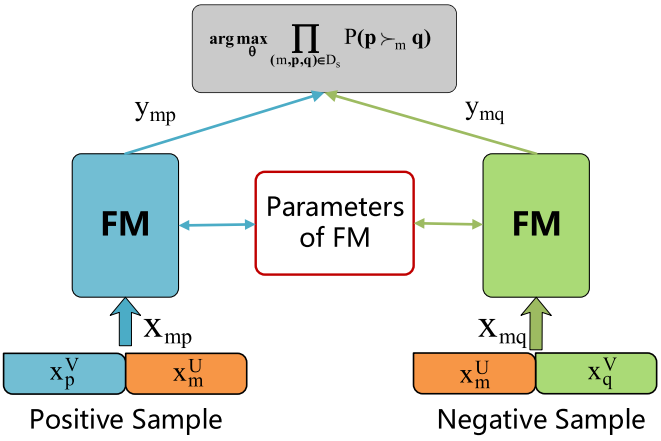


Fig. 1. The illustration of PRFM. The score y_{mp} indicates the degree of user m interested in item p he/she selected, while y_{mq} is the degree of user m interested in item q he/she unselected. y_{mp} and y_{mp} are provided by two FM which share the same parameters. x_m^U, x_p^V and x_q^V are the feature vectors of user m, item p and item q. x_{mp} and x_{mq} are the input features of user-item pairs (m,p) and (m,p) respectively. The preference of user m is defined by $P(p\succ_m q)$.

$$L = \frac{\lambda_f}{2}\|\theta\|^2 + \sum_{(m,p,q)\in D_s}\Big[-\ln \alpha_{mpq}\delta(y_{mp}-y_{mq}) + \sum_{r\in(p,q)} l(\hat{y}_{mr}(x_{mr}|\Theta), y_{mr})\Big] + \frac{\lambda_u}{2}\|U\|^2 + \frac{\lambda_v}{2}\|V\|^2, \tag{7}$$

where $\alpha_{mpq} = \begin{cases} 1+\delta(\hat{y}_{mp}-\hat{y}_{mq}) & \text{if } \hat{y}_{mp} > \hat{y}_{mq} \\ 1-\delta(\hat{y}_{mp}-\hat{y}_{mq}) & \text{else} \end{cases}$. $U \in \mathbb{R}^{M\times z}$ is the low rank matrix of users and $V \in \mathbb{R}^{N\times z}$ is the low rank matrix of items. $l(\hat{y}_{mr}(x_{mr}|\Theta), y_{mr}) = \|\hat{y}_{mr} - y_{mr}\|^2$ is the regression least-squares loss. $\hat{y}_{mr}$ is the preference score of user–item feedback (m, r) predicted by FM, and $y_{mr} = u_m v_r^T$ is the dot product of user latent vector u_m and item latent vector v_r. u_m is the m-th row of U, and v_r is the rth row of V.

4.3. Model fitting

To estimate the parameters of PRFM, i.e., Θ, U and V, we design an alternative optimization algorithm, which uses SGD with drawn training triples to learn U, V and Θ alternatively.

In each iteration, when we are estimating the low rank matrix U or V, we set the parameters of FM, i.e., Θ, to be constants. Then, the gradient of an arbitrary latent parameter $\varepsilon \in (U \vee V)$ is

$$\frac{\partial L}{\partial \varepsilon} = \sum_{(m,p,q)\in D_s} \alpha_{mpq}\left(\delta\left(y_{mp}-y_{mq}\right)-1\right)\frac{\partial\left(y_{mp}-y_{mq}\right)}{\partial\varepsilon} + \lambda_f \frac{\partial \sum_{r\in(p,q)} l(\hat{y}_{mr}(x_{mr}|\Theta), y_{mr})}{\partial\varepsilon} + \lambda\varepsilon, \tag{8}$$

where, λ_f and λ are the hyper-parameters to tune the regularization terms. On the other hand, when we are learning Θ, we set the low rank matrices U and V to be constants. Thus, for an arbitrary strength of feature $w \in \Theta$, its gradient can be calculated by

$$\frac{\partial L}{\partial w} = \begin{cases} \sum_{r\in(p,q)} \hat{y}_{mr} - y_{mr} & \text{if } w \text{ is } w_0 \\ \sum_{r\in(p,q)} (\hat{y}_{mr} - y_{mr}) x_{mri} & \text{if } w \text{ is } w_i \end{cases}. \tag{9}$$

Finally, the gradient of an arbitrary interaction factor $v_{ik} \in \Theta$ can be calculated by

$$\frac{\partial L}{\partial v_{ik}} = \sum_{r\in(p,q)} (\hat{y}_{mr} - y_{mr})\left(x_{mri}\left(\sum_{f=1}^{n} v_{fk} x_{mrf}\right) - v_{ik} x_{mri}^2\right). \tag{10}$$

5. Adaptive sampling

The BPR assumption treats any unselected items as the corresponding users' negative items, and randomly draws massive training triples from unselected items. However, real-world applications just need to select several items from candidate set for a specific user. That causes the users may miss most of items in systems, and the unselected items are not absolutely equal to the negative items. Most of the randomly drawn item pairs are based on BPR assumption, their items hardly appear in front of users at the same time. Making the training triples with such items is neither reasonable nor fair. Moreover, the these randomly drawn triples often make our model converge slowly.

In this section, aim to obtain an efficient learning procedure, we take both content information and implicit feedbacks into consideration, and develop an adaptive item sampler for constructing the training triples. Our strategy automatically approximates realistic distribution of a dataset and adaptively draws informative training triples with a self-evident rule, i.e., draw the item pairs which have a high probability to view by corresponding users.

5.1. Overview

In many real-world applications, items are arranged into different categories. Users often assess the items which are under the same category, and then make their options. Based on this observation, we improve the limitation of BPR assumption with a self-evident rule, i.e., "apples to apples". Specifically, given a positive user-item feedback (u_m, v_p), we assume that its negative item v_q should be comparable to the item v_p as well as have been viewed by user u_m with the item v_p together. Therefore, we design an adaptive sampling strategy which conducts the following steps to select a reasonable item v_q to consist a negative user-item feedback:

- By making use of the feature vector of an positive user–item feedback and the current parameters of FM, we first ascribe the positive feedback (u_m, v_p) to an appropriate category.
- Under a given category, we further select an item v_q which has a high probability to be browsed but not selected by the user u_m.

5.2. Category inference

Given a feature vector $\mathbf{x} = [\mathbf{x}^u, \mathbf{x}^v]$, where $\mathbf{x}^u$ and $\mathbf{x}^v$ are corresponding feature vectors of a user and an item respectively, we want to backward reason about corresponding feedback that the user selected the item probably happened under which categories.

Intuitively, the category distribution of items follows a power law [9], and the features of an item can indicate its categories. Given item v_p, we define the probability of item v_p belonging the dth category as

$$p(d|\mathbf{x}_p^v) \propto \exp\left(\frac{\mathbf{x}_{pd}^v - \mu}{\sigma}\right), \tag{11}$$

where $\mu = E(\mathbf{X}_{*,d}^v)$ and $\sigma = Var(\mathbf{X}_{*,d}^v)$ denote the empirical mean and the variance over all dth features of items, respectively.

Besides items, we also can infer the category of a user-item feedback (u_m, v_p) based on the content information. Specifically,

let $\mathbf{X}^e = [\mathbf{x}_1^e, \mathbf{x}_2^e, \mathbf{x}_3^e, \ldots]$ denote the design matrix of entities (users or items), and $\mathbf{x}_i^e$ is the feature vector of entity e_i. For clarity, we use a superscript u or v to denote a variable associating with users or items. For example, $\mathbf{x}_m^u$ denotes the feature vector of user u_m and $\mathbf{X}^u$ denotes the design matrix of users. We can calculate each weight of categories for the user-item feedback (u_m, v_p) by

$$c_d = \mathbf{x}_{mpd}^v w_d^v + \sum_{j=1}^{N^u} \langle \mathbf{v}_d^v, \mathbf{v}_j^u \rangle \mathbf{x}_{mpd}^v \mathbf{x}_{mpj}^u, \tag{12}$$

where c_d is the weight of the dth category, and N^u is the dimensionality of the user's feature vector. $w_d{}^v$ denotes the strength of the dth feature of item. Then, based on the categorical weights $\mathbf{c} = [c_1, \ldots, c_n]$ of the feedback (u_m, v_p), the categorical distribution of this feedback can be defined as

$$p(d|\mathbf{c}) \propto \exp\left(\frac{\mathbf{c}_d - \mu}{\sigma}\right), \tag{13}$$

where μ and σ denote the empirical mean and the variance over the weight vector $\mathbf{c}$, respectively.

Alternatively, using the categorical distributions of positive feedbacks, we can adaptively sample categories for positive feedbacks, and then choose appropriate negative items under specific categories according to the categorical distributions of items.

5.3. Rank of candidates

Under a given category d, we aim to select an item v_q, which has great potential to have been browsed but not selected by the user u_m. Simply, we can treat the probability $p(d \mid \mathbf{x}_q^v)$ as the ranking score of v_q under the category d, and directly draw items according to their ranking scores. However, there exists a gap between browsing probabilities and ranking scores.

In real-world applications, items are presented in ranking lists, and the top-ranked items usually have significantly high probabilities to be browsed by users than other items. For example, the top three items in a ranking list have much higher probabilities to be browsed by users than the remaining ones, while the ranking scores of items may only have tiny differences. For bridging this gap, we segment the sampling procedure under a given category into two steps. Specifically, we first sample a ranking place r for the candidate from an empirical distribution. Then, we sort items under the category d, and return the item at place r to be a negative item for the positive user-item feedback.

Typically, the empirical distribution of users clicking item on a ranking list approximately follows an analytical law, e.g., a Geometric [25] or Zipf [26] distribution. Here, similar to the work in [9], we adopt the Geometric distribution to draw an item v_q from a ranking list:

$$p(v_q) \propto \exp(-r(q)/\lambda) \quad \lambda \in \mathbb{R}^+, \tag{14}$$

where $r(q)$ denotes the ranking place of item v_q, and λ is a hyper-parameter which tunes the probability density.

Algorithm 1. Adaptive sampling.

Input:
The user–item feedback set D;
The design matrices $\mathbf{X}^u$ and $\mathbf{X}^v$;
The orders of items $\mathbf{L} = \{l_1, l_2, \ldots, l_k\}$;
Output:
The training triple (u_m, v_p, v_q);
1: Draw a rank r from $p(r) \propto \exp(-r/\lambda)$, $(1 \le r \le N)$;
2: Draw $(u_m, v_p) \in D$ randomly;
3: Calculate categorical weights $\mathbf{c}$ for (u_m, v_p) with Eq. (12);
4: Calculate categorical distribution for (u_m, v_p) with Eq. (13);
5: Draw a category d from $p(d|\mathbf{c})$, $(1 \le d \le k)$;
6: $v_q \leftarrow \begin{cases} index(d, r) & \text{if } sgn(\mathbf{X}_{pd}^v) = 1, \\ index(d, n - r - 1) & \text{else} \end{cases}$

5.4. Adaptive sampling algorithm

Theoretically, the sampler for drawing the negative item v_q can be implemented by the following steps. First, sample a ranking position r from the Geometric distribution in $O(\mathbf{1})$. Then, sample a category d from the categorical distribution of the positive feedback (u_m, v_p) in $O(nm)$, where n is the dimensionality of item feature vector, and m is the dimensionality of user feature vector. Finally, sort all candidate items according to their probability of belonging to the category d in $O(|\mathbf{I}| \log |\mathbf{I}|)$, where $|\mathbf{I}|$ is the number of items, and return the item v_q on the rth position. Consequently, for each positive user-item feedback, this algorithm has a complexity of $O(|\mathbf{I}| \log(|\mathbf{I}| + nm + 1))$. This is clearly not feasible in practice, because it is too time consuming.

In fact, we do not need to resort items, since we can precalculate the categorical distribution for each item, and prepare the ranking lists before running the sampling step. On the other hand, if the dimensionality of item feature vector or user feature vector is too high, for a reasonable compromise, we can precompute the categorical distributions of positive user–item feedbacks and update them after each fixed iterations. Because, in each iteration, the parameters of FM change only a little, and after many updating steps it is necessary to change the categorical distributions considerably.

In summary, we propose an adaptive sampling strategy in Algorithm 1 which selects a negative item for a user–item feedback from candidate set. In Algorithm 1, $index(d, r)$ returns the item at place r from the ranking list $l_d \in \mathbf{L}$.

5.5. Adaptive Pairwise Ranking Factorization Machines

For further improving the performance of PRFM in terms of convergence and ranking task, we embed the adaptive sampling strategy into the training procedure of PRFM and summarize the learning algorithm of PRFM in Algorithm 2. It mainly repeats altering learning steps with adaptively drawn data until the parameters reach convergence, where η is the learning rate. $\mathbf{w}^u$ and $\mathbf{w}^v$ are the strength vectors of user and item, respectively. $\mathbf{V}^u$ and $\mathbf{V}^v$ are feature interaction matrices of user and item.

Algorithm 2. Learning parameters for PRFM.

Input:
The user–item feedback set D_S;
The design matrices $\mathbf{X}^u$ and $\mathbf{X}^v$;
Output:
$\Theta := \{w_0, \mathbf{w}^U, \mathbf{w}^V, \mathbf{V}^U, \mathbf{V}^V\}$, U and V;
1: Initialize $\mathbf{V}^U$, $\mathbf{V}^V$, U and V with normal $(0, 1)$;
2: Initialize $\mathbf{w}^U$, $\mathbf{w}^V$ and w_0 with 0;
3: Calculate categorical weights of items by Eq. (11);
4: Sort items with their categorical weights;
5: **repeat**
6: Draw a triple (m, p, q) with Algorithm 1;
7: **for** each latent vector $\varepsilon \in U \vee V$ **do**
8: $\varepsilon \leftarrow \varepsilon + \eta \frac{\partial L}{\partial \varepsilon}$;
9: **end for**
10: **for** each strength of feature $w_i \in \Theta$ **do**
11: $w_i \leftarrow w_i + \eta\left(\frac{\partial L}{\partial w_i} + \lambda w\right)$;
12: **end for**

13: **for** each interaction factor $v_{ik} \in \Theta$ **do**
14: $v_{ik} \leftarrow v_{ik} + \eta\left(\frac{\partial L}{\partial v_{ik}} + \lambda v_{ik}\right)$;
15: **end for**
16: **until** convergence

6. Experiment

In this section, we validate our model by several fundamental experiments. First, we investigate the convergence rate of PRFM. Then, we evaluate the ranking quality of PRFM by Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). Finally, in order to study the cold start problem, we simulate the situation of the cold start and conduct the corresponding comparison experiment.

6.1. Datasets and tasks

We use a real-world dataset for experiments, i.e., MovieLens.[1]

MovieLens includes 100,000 ratings with 943 users and 1682 movies. Each user rated at least 20 movies with the rating values scale 1–5. In experiments, the occupational description, gender and age of a user are treated as content information of the user. Title, genres and published time of a movie are viewed as content information of the movie. Since we mainly study the implicit feedback, we follow the processing method mentioned in [27] to deal with the rating data, i.e., we do not use rating values but just binary rating events by assuming that users tend to rate watched movies. Thus, for a specific user, our task is to predict the potential ranking list of movies.

Fig. 2 shows the statistical characteristics about our experimental dataset. Both users and movies, their in-degree distributions exist a long-tail effect. That is to say, most of the movies only attract a few users, while a very few of movies have been reviewed by most of users. Moreover, comparing users and items, we can observe that the in-degree distribution of items presents a heavier long-tailed effect than that of users.

6.2. Experimental design

To investigate the sampling strategy and the proposed model, we compare our method with state-of-the-arts, and measure their performance by several convictive metrics.

6.2.1. Baselines

MF [28] is a classical factorization method, which is commonly used to deal with collaborative information.

BPR-MF [6] is a matrix factorization method derived under the BPR learning framework. It utilizes the randomly sampling strategy to obtain training data.

Map-BPR [7] is a work aiming for alleviating the cold start problem in original BPR, which independently deals with implicit feedbacks and contents.

FM [2] is a general framework, which can be used to deal with both content information and collaborative information. Here, we use FM to deal with content information of user-item pairs.

RankFM [23] is a recent work aiming at improving FM for dealing with the ranking task.

Table 1 summarizes the characteristics of these compared methods.

[1] http://grouplens.org/datasets/movielens

6.2.2. Experimental setting

In our experiments, we view a user-item pair which does not occur in the implicit feedback dataset as a negative feedback. Thus, the number of negative feedbacks is far more than that of positive feedbacks.

Since both FM and MF treat an implicit feedback as a rating. If we directly use such data to learn FM or MF, then the trained model tends to predict any feedbacks to be the negative feedbacks. Obviously, the trained model is easy to suffer from bad performances on testing datasets. To keep the experimental fair, we train FM and MF with the datasets which contain randomly drawn negative feedbacks. And the number of negative feedbacks is account for 50% in training dataset, because the pairwise models, such as BPR-MF, Map-BPR, RankFM, etc., the number of negative feedbacks in their training dataset can be viewed as 50% of total.

To examine the performance with different dimensionalities ($K = 10, 20, 30, 40, 50$), we carry out training and testing on randomly split training (80%) and testing (20%) data. Moreover, in order to study the cold start problem, we use different amounts of data (20–80%) as the training set and the remaining data (80–20%) as the testing set to mimic the circumstance of cold start and examine the robustness of our model.

In the training phase, we initialize all models with a uniform distribution in all the experiments. And, for evaluating the convergence of compared models, we learn the user/item latent vectors with a same learning rate 0.002, and learn other parameters, such as the mapping matrix of Map-BPR and the parameters of FM with the learning rate 0.0001. Additionally, in BPR-MF, Map-BPR and PRFM, we use three different regularization constants: $\lambda_u = 0.001$ for the user latent vectors, $\lambda_p = 0.002$ for positive updates of item latent vectors, and $\lambda_n = 0.003$ for negative updates of item latent vectors. We give a same regularization constant $\lambda_f = 0.0001$ on the parameters of FM, i.e., the global bias, the strengths of features and the interaction factors of features.

6.2.3. Metrics

For evaluating the ranking performance of models, we follow the evaluation methodology used in [6,7], and report the value of AUC (the area under the ROC curve)

$$AUC = \frac{1}{M_{test}} \sum_{u} \frac{1}{|E(u)|} \sum_{(p,q)\in E(u)} \Phi(x_{mpq}),$$

where M_{test} is the number of testing users, $E(u) := \{(p, q) | (u, p) \in D_{test} \wedge (u, q) \notin D_{test} \wedge D_{train}\}$. AUC is a popular indicator which is used to measured the quality of personalized ranking with implicit feedbacks. A higher value of the AUC indicates a better ranking quality. In our experiments, the best achievable AUC is 1, while the trivial AUC of a random guess is 0.5.

Besides, the Normalized Discounted Cumulative Gain (NDCG) is a common measure of ranking quality in information retrieval, which can measure the usefulness, or gain, of an item based on its position in the ranking list. Naturally, in this work, we use NDCG to measure the performance of Top-N ranking. NDCG is defined as

$$NDCG(N) = \frac{1}{Z_N} \sum_{i=1}^{N} \frac{2^{r(i)} - 1}{\log(i+1)},$$

where N is the truncated position in a ranking list, Z_N is a normalization constant to ensure that the NDCG score for the correct ranking is one, and $r(i)$ is the graded relevance of the ith point in the ranking list.

6.3. Results and analysis

In this subsection, we investigate our model from various views. First, we analyze convergence of our model. Then, we

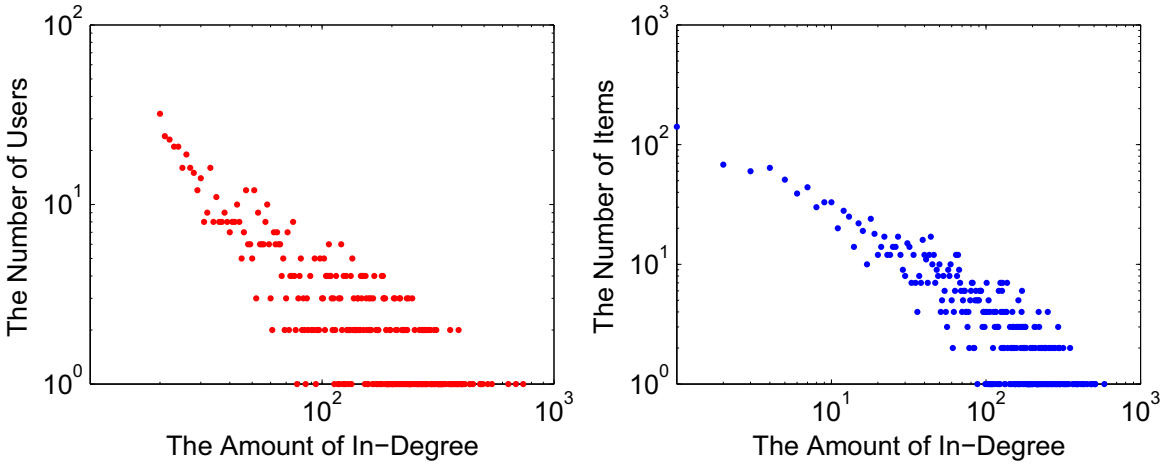


Fig. 2. Degree distributions in the MovieLens dataset. The plots show in-degree distributions of user and movie respectively on a log–log scale.

Table 1
Characteristics of compared methods.

Method	Content	Feedback	Sampling
MF	No	Implicit	–
FM	Yes	Implicit/explicit	–
BPR-MF	No	Implicit	Random
Map-BPR	Yes	Implicit	Random
RankFM	Yes	Implicit	Random

evaluate the ranking quality of comparing models by AUC and NDCG respectively. Finally, we test our model in the simulative cold start circumstance.

6.3.1. *Learning efficiency*

Fig. 3 shows the convergence comparisons on MovieLens dataset, where the performance of methods is evaluated by AUC in each training epoch. Fig. 3(a) shows the convergence rates of our model with different dimensionalities. As the increase of iterations, the AUC value of our model with various dimensionalities increases greatly at the beginning, and then changes slightly after around 200 iterations. Since our sampling strategy can well draw the training data, when the number of dimensions is increasing, the convergence rates of our model are also increasing. From Fig. 3 (b), we can observe that PRFM has improvement both on convergence rate and ranking quality comparing with other pairwise methods, i.e., BPR-MF, RankFM and Map-BPR. RankFM significantly converges slower than other methods. Map-BPR and BPR-MF have similar convergence rates, since they adopt the same sampling strategy. By incorporating content information of users and items into the sampling strategy, our model can achieve a better ranking performance in term of AUC, and converge faster than compared methods.

Fig. 4 gives the average running-time of different methods in one training epoch. Obviously, among of methods, PRFM, Map-BPR and RankFM, which are able to utilize the content information, PRFM is the most efficient one because of embedding the proposed adaptive sampler. Since BPR-MF does not consider the content information, it costs the minimum running-time in each training epoch. However, as indicated in Fig. 3(a), the BPR-MF usually needs more iterative epochs and achieves worse predictive performance than PRFM, on account of using the random sampler and without considering the content information.

6.3.2. *Ranking quality*

Table 2 gives the performance of Top-N ($N = 3, 5, 10, 20$) ranking with various dimensionalities. Due to well incorporating content information and implicit feedbacks into a unified framework, our model outperforms other methods in most of dimensionalities, and get the best results in all Top-N ranking tasks. Specifically, in each Top-N ranking task, our model can increase the performance of the suboptimal method 7–13%. Besides, since

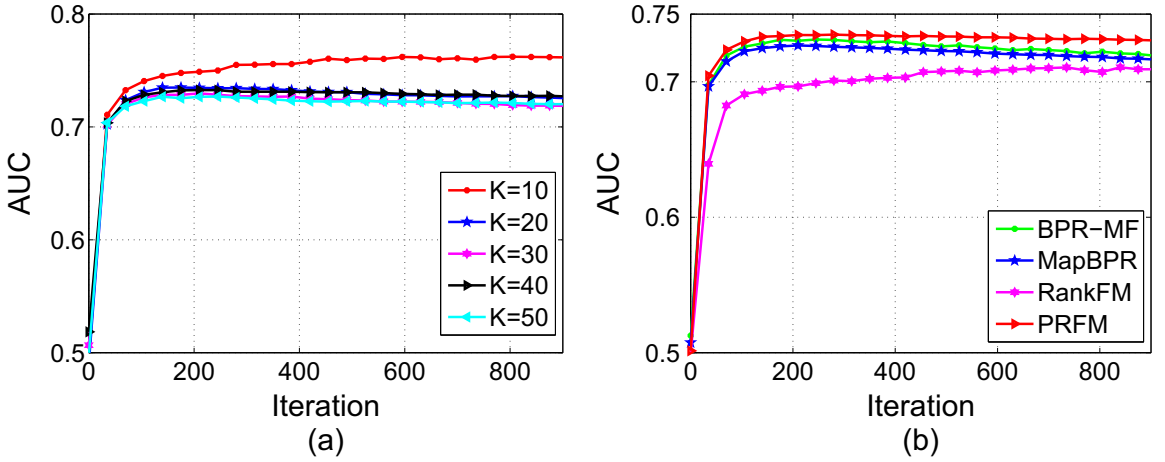


Fig. 3. Convergence comparison on MovieLens. The horizontal axis is the number of training epoch, and the vertical axis indicates the values of AUC. The plot (a) indicates convergence rates of PRFM with different dimensions. The plot (b) shows the convergence rates of PRFM and other pairwise methods, where each line represents a specific model and their performance of AUC are measured by the mean of different dimensionalities ($K = 10, 20, 30, 40, 50$).

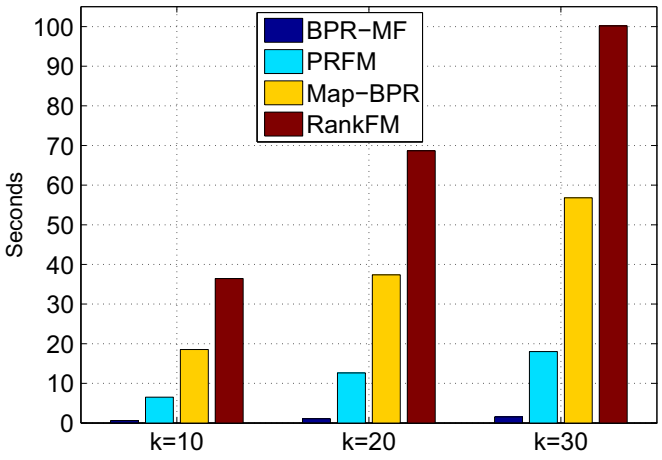


Fig. 4. The running-time comparison on MovieLens. Each bar indicates the average training time of a specific method with given dimensionalities (K = 10, 20, 30) in one training epoch.

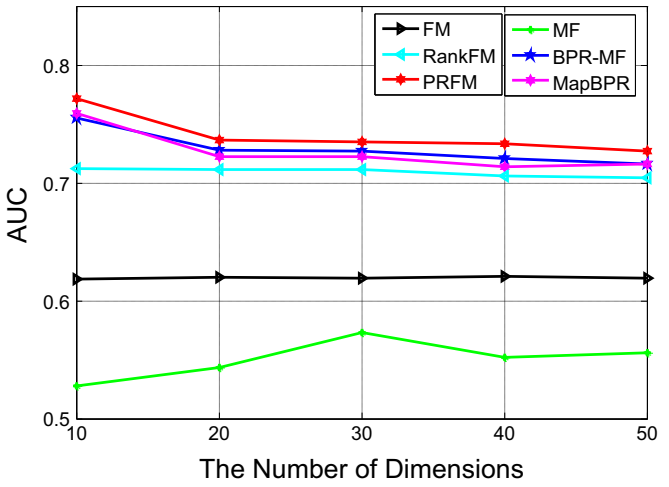


Fig. 5. The ranking performance evaluated by AUC on the MovieLens with different dimensionalities.

implicit feedbacks or content information cannot express the user preference on different items directly, both FM and MF which do not take account of such ranking clues have bad performance on Top-N ranking.

Moreover, we investigate the ranking performance of compared methods with the criterion AUC. Fig. 5 shows the ranking performance of different methods with various dimensionalities. Owing much to combining implicit feedbacks with content information, PRFM consistently outperforms other methods. As well as the results in Table 2, the performance of MF and FM is worse than other methods, since they only take either implicit feedbacks or content information. By including ranking clues which are derived by the pairwise assumption of BPR, BPR-MF is able to obtain better performance than MF, FM and even RankFM.

Additionally, RankFM which incorporates the ranking clues and content information with FM by making use of BPR-OPT criterion indeed improves the FM in the ranking task.

6.3.3. Cold start

For investigating the cold start problem, we mimic the cold start situation by training and testing content aware models, i.e., PRFM, Map-BPR, FM and RankFM, with different amounts of training data (80–20%) and testing data (20–80%).

Table 3 reports the performance of models, which are trained by different amounts of data with K=40, on Top-N ranking. Generally, with the increase of training data, most of the models have improvements in all Top-N ranking tasks. Specifically, due to well incorporating content information with implicit feedbacks, our model avoids the dramatic fluctuation of performance with the amount of training data. Moreover, PRFM can get the best results, except for using 40% data to train and conducting the Top-10 and Top-20 ranking tasks on 60% data.

Furthermore, Fig. 6 shows the ranking performance comparison of different models. The dimensionality of models is K=40. Each color of bars represents the performance of a model trained by different amounts of data. Since both user set and item set present long-tail effects, when the training data is not very spare, most of the users and items still appear in the training dataset. Therefore, as the decreasing of training data, the performance of methods which take account of content information is not decreasing dramatically at beginning. Note that when the training data is extremely spare, even the popular entities (users or items) which have enough content information are hard to appear in the training dataset. That is to say, the content aware parameters of the FM and our model are hardly learned very well. That may be the reason why their performance decreases a lot in the case of using 20% data for training and 80% data for testing. Generally speaking, our model outperforms other compared methods in other cases, and can alleviate the cold start problem in the task of personalized ranking.

Table 2
The ranking performance comparison on MovieLens with different dimensionalities (K = 10, 20, 30, 40, 50). The performance is evaluated by NDCG@ (3, 5, 10, 20).

Method	NDCG@3					NDCG@5				
	K=10	K=20	K=30	K=40	K=50	K=10	K=20	K=30	K=40	K=50
MF	0.083	0.090	0.190	0.161	0.121	0.075	0.098	0.207	0.158	0.125
BPR-MF	0.436	0.426	0.390	0.389	0.399	0.453	0.449	**0.451**	0.419	0.439
Map-BPR	0.440	0.429	0.426	0.423	0.453	0.453	0.451	0.444	0.408	0.472
FM	0.186	0.227	0.214	0.222	0.219	0.243	0.262	0.226	0.264	0.249
RankFM	0.388	0.404	0.429	0.425	0.418	0.409	0.414	0.431	0.428	0.412
PRFM	**0.491**	**0.436**	**0.436**	**0.458**	**0.505**	**0.513**	**0.453**	0.442	**0.482**	**0.520**
	NDCG@10					NDCG@20				
MF	0.087	0.138	0.193	0.152	0.132	0.097	0.181	0.218	0.186	0.133
BPR-MF	0.460	**0.465**	0.421	0.456	0.463	0.461	**0.455**	0.425	0.460	0.475
Map-BPR	0.477	0.448	0.466	0.427	0.505	0.480	0.447	0.449	0.427	0.497
FM	0.271	0.306	0.258	0.284	0.274	0.315	0.342	0.303	0.306	0.317
RankFM	0.400	0.443	0.424	0.431	0.431	0.412	0.420	0.431	0.417	0.428
PRFM	**0.506**	0.460	**0.481**	**0.484**	**0.529**	**0.504**	0.451	**0.475**	**0.478**	**0.533**

Table 3
The ranking performance comparison of different methods on MovieLens with different amount of training data (20–70%). The performance is evaluated by NDCG@(3,10,20) when $K=50$.

Method	NDCG@3				NDCG@10				NDCG@20			
	20%	40%	60%	70%	20%	40%	60%	70%	20%	40%	60%	70%
Map-BPR	0.401	0.482	0.472	0.419	0.486	0.482	0.517	0.434	0.489	0.486	0.507	0.454
FM	0.328	0.221	0.284	0.252	0.460	0.286	0.349	0.311	0.452	0.298	0.353	0.323
RankFM	0.461	0.462	0.463	0.433	0.524	**0.534**	0.475	0.457	0.502	**0.524**	0.490	0.467
PRFM	**0.504**	**0.528**	**0.493**	**0.515**	**0.521**	0.481	**0.527**	**0.530**	**0.525**	0.499	**0.526**	**0.529**

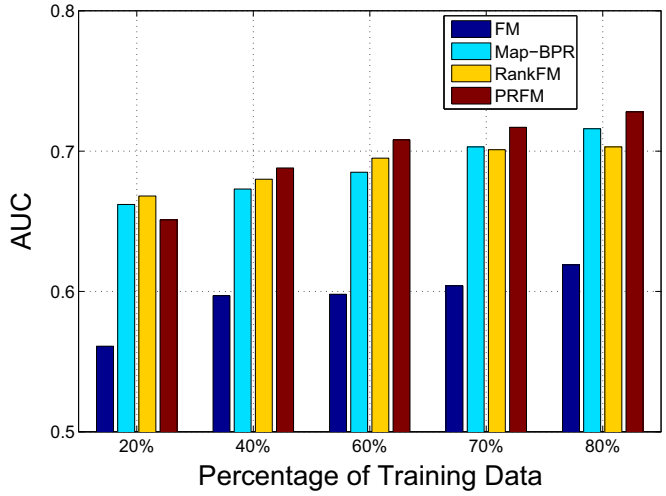


Fig. 6. Performance comparison of different methods on MovieLens with different amounts of training data. The performance of different models is measured by AUC, where the dimensionality of models is set as $K=50$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

7. Conclusion

Traditional pairwise learning algorithms for personalized ranking usually have two fundamental limitations, i.e., long training process caused by the randomly sampling strategy and the neglect of content information. In this work, we have illustrated our solution for dealing with these limitations. Specifically, we have constructed a novel model, PRFM (Pairwise Ranking Factorization Machines), which incorporates implicit feedbacks with content information for the task of personalized ranking. To speed up the learning procedure, we have developed an adaptive sampling strategy, which utilizes content information of user and item to obtain a better balance of efficiency and performance. Through comprehensive experiments on a real-world dataset, we have demonstrated that our solution outperforms previous methods on the task of personalized ranking, and can well alleviate the cold start problem.

In real-world applications, content information usually include multiple types, noises and redundant, as well even the positive feedbacks in implicit feedback datasets often have noises, e.g., random clicks. These deficiencies of data may lead to performance degeneration. In the future, for obtaining more robust solution, we plan to elaborately incorporate the ability of processing sparse contents into our model, and improve our adaptive sampler by exploring negative items under multiple categories.

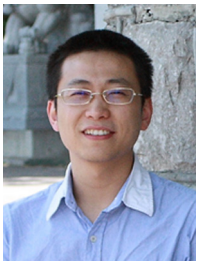
Acknowledgement

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

References

[1] Y. Shi, M. Larson, A. Hanjalic, List-wise learning to rank with matrix factorization for collaborative filtering, in: RecSys, 2010, pp. 269–272.
[2] S. Rendle, Factorization machines with libFM, TIST 3 (3) (2012) 57:1–57:22.
[3] S. Qiu, J. Cheng, T. Yuan, C. Leng, H. Lu, Item group based pairwise preference learning for personalized ranking, in: SIGIR, 2014, pp. 1219–1222.
[4] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: ICDL, 2000, pp. 195–204.
[5] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: Recommender Systems Handbook, Springer, 2011, pp. 73–105.
[6] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: UAI, 2009, pp. 452–461.
[7] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, L. Schmidt-Thieme, Learning attribute-to-feature mappings for cold-start recommendations, in: ICDM, 2010, pp. 176–185.
[8] W. Pan, L. Chen, Gbpr: Group preference based Bayesian personalized ranking for one-class collaborative filtering, in: IJCAI, 2013, pp. 2691–2697.
[9] S. Rendle, C. Freudenthaler, Improving pairwise learning for item recommendation from implicit feedback, in: WSDM, 2014, pp. 273–282.
[10] H. Zhong, W. Pan, C. Xu, Z. Yin, Z. Ming, Adaptive pairwise preference learning for collaborative recommendation with implicit feedbacks, in: CIKM, 2014, pp. 1999–2002.
[11] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: ICDM, 2008, pp. 263–272.
[12] L. Hong, A.S. Doumith, B.D. Davison, Co-factorization machines: modeling user interests and predicting individual decisions in twitter, in: WSDM, 2013, pp. 557–566.
[13] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: ICDM, 2011, pp. 287–296.
[14] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inf. Retr. 4 (2) (2001) 133–151.
[15] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Incremental singular value decomposition algorithms for highly scalable recommender systems, in: ICCIS, 2002, pp. 27–28.
[16] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: ICDM, 2008, pp. 426–434.
[17] Y. Li, J. Hu, C. Zhai, Y. Chen, Improving one-class collaborative filtering by incorporating rich user information, in: CIKM, 2010, pp. 959–968.
[18] R. Gemulla, E. Nijkamp, P. J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, in: SIGKDD, 2011, pp. 69–77.
[19] W. Zhang, H. Sun, X. Liu, X. Guo, Temporal qos-aware web service recommendation via non-negative tensor factorization, in: WWW, 2014, pp. 585–596.
[20] G. Weiyu, W. Shu, W. Liang, T. Tieniu, Adaptive pairwise learning for personalized ranking with content and implicit feedback, in: WI, 2015.
[21] P.W. Foltz, S.T. Dumais, Personalized information delivery: an analysis of information filtering methods, Commun. ACM 35 (12) (1992) 51–60.
[22] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: Recommender Systems Handbook, 2011, pp. 73–105.
[23] I. Bayer, fastfm: A library for factorization machines, arXiv preprint arXiv:1505.00641.
[24] W. Guo, S. Wu, L. Wang, T. Tan, Multiple attribute aware personalized ranking, in: Web Technologies and Applications, 2015, pp. 244–255.
[25] K. Wang, T. Walker, Z. Zheng, Pskip: estimating relevance ranking quality from web search clickthrough data, in: SIGKDD, 2009, pp. 1355–1364.
[26] L.A. Adamic, B.A. Huberman, Zipfs law and the Internet, Glottometrics 3 (1) (2002) 143–150.
[27] G. et al. A unified approach to building hybrid recommender systems, in: RecSys, 2009, pp. 117–124.
[28] W.S. Chin, Y. Zhuang, Y.C. Juan, C.J. Lin, A learning-rate schedule for stochastic gradient methods to matrix factorization, in: PAKDD, 2015, pp. 442–455.

Weiyu Guo received the B.E. degree in School of Computer and Information Technology from Shanxi University, China, in 2010, the M.E. degree in Software Institute from Nanjing University, China, in 2012, and the PhD degree from the School of Engineering Science, University of Chinese Academy of Sciences, in 2016. His research interests include machine learning, recommendation system and large-scale Web data mining.



Shu Wu received the B.E. degree from Hunan University, China, in 2004, the M.E. degree from Xiamen University, China, in 2007, and the PhD degree from the University of Sherbrooke, Canada, in 2012, all in computer science. He is an assistant professor in the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. He has published more than 20 papers in the areas of data mining and information retrieval at international journals and conferences, such as IEEE TKDE, AAAI, SIGIR, CIKM. His research interests include data mining, recommendation systems, and pervasive computing. He is a member of the IEEE.

Liang Wang received both the B.E. and M.E. degrees from Anhui University in 1997 and 2000 respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA) in 2004. From 2004 to 2010, he has been working as a Research Assistant at Imperial College London, United Kingdom and Monash University, Australia, a Research Fellow at the University of Melbourne, Australia, and a lecturer at the University of Bath, United Kingdom, respectively. Currently, he is a full Professor of Hundred Talents Program at the National Lab of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition, computer vision and data mining. He has widely published at highly ranked international journals such as IEEE TKDE, IEEE TPAMI and IEEE TIP, and leading international conferences such as ICDM, CVPR, and ICCV. He is an associate editor of IEEE Transactions on SMC-B, International Journal of Image and Graphics, Signal Processing, Neurocomputing and International Journal of Cognitive Biometrics. He is currently a Senior Member of IEEE.

Tieniu Tan received his BSc degree in electronic engineering from Xi'an Jiaotong University, China, in 1984, and his MSc and PhD degrees in electronic engineering from Imperial College London, U.K., in 1986 and 1989, respectively. In October 1989, he joined the Computational Vision Group at the Department of Computer Science, The University of Reading, Reading, U.K., where he worked as a Research Fellow, Senior Research Fellow and Lecturer. In January 1998, he returned to China to join the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of the Chinese Academy of Sciences (CAS), Beijing, China, where he is currently Professor and former director (1998–2013) of the NLPR and Center for Research on Intelligent Perception and Computing (CRIPAC), and was Director General of the Institute (2000–2007). He is currently also Deputy Secretary-General of the Chinese Academy of Sciences. He has published 11 edited books or monographs and more than 400 research papers in refereed international journals and conferences in the areas of image processing, computer vision and pattern recognition. His H-index is 61 (as of December 2014). His current research interests include biometrics, image and video understanding, and information content security.

Dr Tan is a Fellow of CAS, TWAS (The World Academy of Sciences for the advancement of science in developing countries), IEEE and IAPR (the International Association of Pattern Recognition), and an International Fellow of the UK Royal Academy of Engineering. He has served as chair or program committee member for many major national and international conferences. He is or has served as an associate editor or a member of editorial boards of many leading international journals including IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Circuits and Systems for Video Technology, Pattern Recognition, Pattern Recognition Letters, Image and Vision Computing, etc. He is Editor-in-Chief of the International Journal of Automation and Computing. He was founding chair of the IAPR Technical Committee on Biometrics, the IAPR-IEEE International Conference on Biometrics, the IEEE International Workshop on Visual Surveillance and Asian Conference on Pattern Recognition (ACPR). He currently serves as President of the IEEE Biometrics Council and Deputy President of Chinese Artificial Intelligence Association. He has given invited talks and keynotes at many universities and international conferences, and has received many national and international awards and recognitions.