User Preference Learning for Online Social Recommendation

Zhou Zhao, Deng Cai, Xiaofei He and Yueting Zhuang

Abstract—Social recommendation system has attracted a lot of attention recently in the research communities of information retrieval, machine learning and data mining. Traditional social recommendation algorithms are often based on batch machine learning methods which suffer from several critical limitations, e.g., extremely expensive model retraining cost whenever new user ratings arrive, unable to capture the change of user preferences over time. Therefore, it is important to make social recommendation system suitable for real-world online applications where data often arrives sequentially and user preferences may change dynamically and rapidly. In this paper, we present a new framework of online social recommendation from the viewpoint of online graph regularized user preference learning (OGRPL), which incorporates both collaborative user-item relationship as well as item content features into an unified preference learning process. We further develop an efficient iterative procedure, OGRPL-FW which utilizes the Frank-Wolfe algorithm, to solve the proposed online optimization problem. We conduct extensive experiments on several large-scale datasets, in which the encouraging results demonstrate that the proposed algorithms obtain significantly lower errors (in terms of both RMSE and MAE) than the state-of-the-art online recommendation methods when receiving the same amount of training data in the online learning process.

Index Terms—Online social recommendation, user preference learning, low rank

1 INTRODUCTION

With the increasing popularity of social media, social recommendation has attracted a lot of attention recently in the research communities of information retrieval [50], machine learning [57] and data mining [35], due to the potential value of social relations [42], [7], [27]. We have witnessed the many popular commercial social recommender systems such as Douban [27] and Epinions [28].

In literature, a variety of social recommendation models are proposed, which can be generally grouped in two categorizes: matrix factorization based methods and probabilistic model based methods. The methods of both categories are trained from the partially observed user-item matrix and users’ social relations. The matrix factorization based approaches [11], [10], [41] factorize the partially observed user-item matrix into two latent low-rank matrices with the regularization of users’ social relations, and then fill the missing data entries by spanning two low-rank matrices. On the other hand, the probabilistic model based approaches [35], [18], [43] infer the probabilistic model from the partially observed user-item matrix and then predict the missing entries based on the probabilistic model.

Despite the extensive studies of social recommendation systems [11], [10], [41], [35], [18], [43], most traditional social recommendation algorithms are based on batch training techniques which assume all user ratings are provided in the user-item matrix. Such assumption make them unsuitable for real-world online recommendation applications. First, the user ratings arrive sequentially in an online applications. The batch recommendation algorithm has to be retrained from scratch whenever new ratings are received, making the training process extremely time-consuming. Moreover, if the size of training data is too large, it is difficult for handling all the data in the batch mode. Second, it is common that user preference could drift over time in real-world online application, which makes the batch learning processes fail to capture such changes on time.

To overcome these difficulties, we develop a novel framework of social recommender system termed Online Graph Regularized User Preference Learning (OGRPL). In the task of online recommendation, the number of user ratings collected at each timestamp is much smaller than the ratings in the offline recommendation, which means all the items have to be recommended in a cold-start manner. Currently, social networking and knowledge sharing sites like Twitter and Douban are popular platforms for users to generate shared opinions for the items like item review and summary [14]. Thus, the user generated content provides the auxiliary information for the items, which has been widely used to tackle the problem of cold-start item [9], [31]. Unlike the existing online collaborative filtering methods [46], [3], [24], OGRPL is a hybrid model utilizing both CF information via the partially observed user-item matrix as well as the auxiliary content features for each item. Given a stream of user ratings, OGRPL incrementally learns the user preference on the content features of the items. However, humans are prone to make rating errors and the rating data always contain noise in practice. Thus, the direct learning of user preference may be over-fitting and is therefore not robust. To overcome the overfitting problem, we formulate the problem of user preference learning with low-rank constraints and learn the low-rank representation of user
2 Related Work

In this section, we briefly review some related work on the problems of social recommendation and online learning.

2.1 Social Recommendation

The social recommendation models are trained from the partially observed user-item matrix and users’ social relations. Gao et. al. [11] study the point of interest recommendation based on the content information from the location-based social networks. Qian et. al. [35] incorporate the CircleCon model with probabilistic matrix factorization method for social recommendation. Wang et. al. [48] design a joint social-content recommendation framework to suggest users which video to import or re-share in the online social network. Jiang et. al. [18] present the social contextual information based probabilistic matrix factorization for recommendation. Qiao et. al. [36] study the event recommendation by combining both online and offline social networks. Luo et. al. [26] devise the social-based collaborative filtering recommendation using users’ heterogeneous relations. Lu et. al. [25] model the dynamic user interest evolving effect and suggestions made by the recommender instigate an interest cascade over the users. Ding et. al. [6] study the celebrity recommendation based on collaborative social topic regression. Wang et. al. [43] present the tag recommendation based on social regularized collaborative topic regression. Tang et. al. [41] propose the global and local regularization for social recommendation. Gao et. al. [10] study the location recommendation on location-based social networks with temporal constraints. Liu et. al. [21] proposes point of interest recommendation system with topic and location awareness in location-based social networks. Zhang et. al. [53] present the domain-specific recommendation system TopRec, which mines community topic in social networks. Hu et. al. [14] propose a framework MR3 to jointly model ratings, item reviews and social graph for rating prediction. Wang et. al. [44] study the news recommendation in social media. Zhou et. al. [56] study the user recommendation in social tagging systems based on users’ personal interests. Zhao et. al. [55] present the expert finding for question answering in social networks. Perushotham et. al. [34] propose the collaborative topic regression method with social matrix factorization for social recommendation. Yang et. al. [50] propose the socialMF model for recommendation regularized by friend circles. Feng et. al. [8] propose a multi-type graph for social tagging systems which learn the weights of different types of nodes and edges. Shen et. al. [38] develop a joint personal and social latent factor model for social recommendation, which combines collaborative filtering and social network modeling.
approaches. Zhu et. al. [57] formulate the graph Laplacian
regularized social recommendation into a low-rank semidefinite
program, which is solved by the quasi-Newton algorithm. The
survey of existing social recommendation methods can be found in [42], [7], [27].

Unlike the previous studies, we study the problem of online
social recommendation where the user preference is updated
online with graph regularization.

2.2 Online Learning

Online learning methods learn from one or a group of
sample data each time by updating the learning model, which
have been applied to the problems of dictionary learning [29],
[23], [49], [19], [29], feature selection [45], [46], [32] and
and collaborative filtering [37], [3], [12], [30], [46], [1]. In this
section, we mainly review the online learning techniques to
the problem of collaborative filtering below.

Recent years have witnessed some emerging studies for
online collaborative filtering [1], which follow the first order
optimization framework in finding the optimal solutions of
low-rank matrix factorization using online gradient descent.
Blondel et. al. [3] propose the online nonnegative matrix
factorization under the framework of passive-aggressive learning.
Mairal et. al. [30] propose the online optimization algorithm
based on stochastic optimization for nonnegative matrix
factorization. Qiao et. al. [37] present online nonparametric
max-margin matrix factorization for collaborative filtering.
Guan et. al. [12] study online nonnegative matrix factorization
via robust stochastic optimization to update the bases in an
incremental manner.

Unlike the previous studies, we study the problem of online
social recommendation incorporating both collaborative user-
item relationship as well as item content features into an
unified preference learning process.

3 THE PROBLEM OF ONLINE SOCIAL RECOM-
MENDATION

In this section, we first introduce some notations used in
the subsequent discussion, which are the rating matrix \( \mathbf{R} \), the
feature content matrix of items \( \mathbf{X} \), the similarity matrix of
users’ social relations \( \mathbf{S} \) and the target preference matrix \( \mathbf{W} \).
We next present the problem of online social recommendation
from the viewpoint of online graph regularized user preference
learning. We then learn the user preference using both the
collaborative user-item relationship and item content features
as an unified learning process.

We represent the item content feature in recommender
systems using bag-of-words model. We take the item review
by users in the recommender system as the content feature of
the items. The review text contains rich information of items.
We denote each item \( \mathbf{x}_i \) by a \( d \)-dimensional word
vector. We then denote the collection of items by \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{d \times m} \) where \( m \) is the total number of
the items. We denote the collection of users in recommender
systems by \( \mathbf{W} = [w_1, \ldots, w_n] \in \mathbb{R}^{d \times m} \) where \( n \) is the total
number of the users. Given the user preference matrix \( \mathbf{W} \), the
i-th user preference is represented by \( w_i = \mathbf{W}(i) \). The \( w_j \) is
a \( d \)-dimensional vector for modeling the preference of the j-th
user on word feature. The terms in \( w_j \) indicate the preference
strengths on the word feature of the items. For example, in
movie recommendation, consider that the i-th user prefers
comedy movie to action movie, the weight of the k-th word
“comedy” is larger than the l-th word “action” in vector \( w_j \).
That is, \( w_j(k) > w_j(l) \). However, most CF models are built on
latent vectors generated from the matrix factorization, which are
difficult to interpret.

We denote the user-item rating matrix by \( \mathbf{R} \in \mathbb{R}^{n \times m} \).
The value in the matrix \( \mathbf{R} \) is rated by the users in the
recommender systems, which indicates the user preference for
the recommended items. We notice that the user ratings are
sequentially collected and we denote that the collection of user
ratings at the k-th timestamp (or the k-th round) by \( \Omega_k \). We use
the term timestamp and round interchangeable throughout the
paper. We then denote the collection of user ratings from the
1-th round to the K-th round by \( \Omega = \{ \Omega_1, \ldots, \Omega_K \} \).

The rating value of the j-th item by the i-th user exists if its
index \((i,j)\in\Omega_k\) exists at certain round \( k \). We notice that
the matrix \( \mathbf{R} \) is sparse and a number of values are missing in \( \mathbf{R} \).

We now show that the learning of user preference by
incorporating both user-item collaborative relationship and
content feature of the items. We represent the rating prediction
of the users by function \( f_{\mathbf{w}}(\cdot) \) where vector \( \mathbf{w} \) is the user
preference. Thus, the rating prediction of the j-th item by the
i-th user is given by \( \hat{r}_{ij} = f_{\mathbf{w}}(x_j) \) and the prediction
of matrix \( \mathbf{R} \) is given by \( \hat{\mathbf{R}} = f_{\mathbf{W}}(\mathbf{X}) = \mathbf{W}^T \mathbf{X} \) where the user
preference matrix \( \mathbf{W} = [w_1, \ldots, w_n] \). To avoid the overfitting
of the function learning and preserve collaborative relationship
between users and items, we learn the user preference matrix
with low-rank constraint.

On the other hand, the benefits of the potential value of
social relations have been well-recognized in social recom-
mender systems. We present the online user preference
learning framework with graph regularization, which utilizes
the users’ social relations for improving the quality of online
recommendation. We denote the similarity matrix between
users by \( \mathbf{S} \in \mathbb{R}^{n \times n} \). For the Facebook-like social friendship,
the similarity entry \( s_{ij} = 1 \) when the i-th user and the j-th
user are friends, otherwise \( s_{ij} = 0 \). For the Microblogging-
like following relationship, we let \( \mathbf{F}_i \) be the set of following

\[
\begin{array}{|c|c|}
\hline
\text{Notation} & \text{Notation Description} \\
\hline \mathbf{R} & \text{an observed rating matrix} \\
\mathbf{X} & \text{a content feature matrix of items} \\
\mathbf{W} & \text{an coefficient matrix of user preference} \\
\mathbf{S} & \text{a similarity matrix of users} \\
\mathbf{F}_1, \ldots, \mathbf{F}_n & \text{set of following users} \\
\mathbf{L} & \text{a laplacian matrix of users} \\
\mathbf{D} & \text{a diagonal matrix of users} \\
\Omega_k & \text{rating indices at the k-th round} \\
\Omega = \{\Omega_1, \ldots, \Omega_K\} & \text{a sequential collection of existing rating indices} \\
\mathbf{W}_1, \ldots, \mathbf{W}_K & \text{a sequential collection of user preference} \\
\mathbf{I}_\Omega & \text{an indicator matrix for observed ratings} \\
f_{\mathbf{W}}(\mathbf{X}) = \mathbf{W}^T \mathbf{X} & \text{the user preference function} \\
\lambda & \text{a regularization term} \\
\hline
\end{array}
\]
users of the $i$-th user and $F_j$ be the set of following users of the $j$-th user. We use the Jaccard Distance to model the similarity between the $i$-th user and the $j$-th user, which is $s_{ij} = \frac{|F_i \cap F_j|}{|F_i \cup F_j|}$. $|F_i \cap F_j|$ is the set of two users’ common followings and $|F_i \cup F_j|$ is the set of two users’ total followings. We note that the similarity value in $S$ is within the range $[0, 1]$. 

Using the notations above, we define the problem of online social recommendation from the viewpoint of online user preference learning as follows. Given a sequential collection of user ratings from the $1$-th round to the $K$-th round $\Omega = \{\Omega_1, \ldots, \Omega_K\}$, the goal is to learn the user preference matrix $W_1, \ldots, W_K$ sequentially in order to predict the missing values in the user-item rating matrix $R$ online.

### 4 The Objective Function Formulation

In this section, we formulate the objective function for the problem of online social recommendation from the viewpoint of online user preference learning, and then present the objective function in the setting of online learning.

Given a sequential collection of user ratings with indices $\Omega_1, \ldots, \Omega_K$, we aim to provide the online social recommendation without full re-computation of user preference $W_k$ at each round. Consider the collection of user ratings at the $k$-th round $\Omega_k$, the user-item rating matrix $R$ and the user preference at the previous round $W_{k-1}$, we first learn the user preference $W_k = [w_{1,1}, \ldots, w_{n,n}] \in R^{d \times n}$ and then provide the recommendation by predicting the user ratings $\hat R = f_{W_k}(X) = W_k^T X$.

We denote the rating value of the $j$-th item by the $i$-th user by $r_{ij}$. By requiring $r_{ij} = f_{W_k}(x_j) = w_{ij}^T x_j$, we expect to learn the $i$-th user preference $w_i$ from his/her rating on the content feature of the $j$-th item. However, we notice that the rating values in the user-item matrix $R$ may be noisy since users are prone to make errors. Therefore, the hard constraint is not robust to the noise in the user-item rating matrix $R$. To overcome this limitation, we introduce the loss function to relax the hard constraint of user rating prediction, given by

$$l_{W_k}(X, R, \Omega_k) = \frac{1}{|\Omega_k|} \sum_{(i,j) \in \Omega_k, w_i = w_k(i)} (r_{ij} - w_{ij}^T x_j)^2$$

$$= \frac{1}{|\Omega_k|} \|I_{\Omega_k} \odot (R - f_{W_k}(X))\|^2_F,$$

(1)

where $l_{W_k}(X, R, \Omega_k)$ is the loss function of user preference function at the $k$-th round, $\| \cdot \|^2_F$ denotes the Frobenius norm, and $\odot$ represents the Hadamard element-wise product. $I_{\Omega_k}$ is an indicator matrix at the $k$-th round for the observed ratings between users and items, and zeros for the missing values. $|\Omega_k|$ is the number of rating indices at the $k$-th round. We denote the loss function of the sequential collection of user ratings from the $1$-th round to the $K$-th round based on user preference by $\frac{1}{K} \sum_{k=1}^K l_{W_k}(X, R, \Omega_k)$.

Note that, the user preference $W = [w_1, \ldots, w_n]$ may be correlated in the real-world recommendation applications. Therefore, to avoid the overfitting problem of user preference learning, it is natural to assume that the user preference matrix $W$ is of low rank. Consequently, we cast the problem of online social recommendation via online user preference learning into the online low-rank optimization on user preference matrix $W$ (i.e., $\min_W \|W\|_{\text{rank}}$). Unfortunately, the rank minimization problem is NP-hard in general due to the nonconvexity and discontinuous nature of the rank function. We follow the recent studies [17], [51], [52] on the approximation of rank minimization using trace norm, which is a convex surrogate of the non-convex matrix rank function. We then give objective function of low-rank user preference learning in the form of bounding trace norm [16] at the $K$-th round, given by

$$\min_{W_K} \mathcal{F}_K(W_K) = \frac{1}{K} \sum_{\tau=1}^K l_{W_K}(X, R, \Omega_{\tau})$$

$$\text{s.t.} \quad \|W_K\|_* \leq \gamma,$$

(2)

where $\sum_{\tau=1}^K l_{W_K}(X, R, \Omega_{\tau})$ is the total loss function from the $1$-th round to the $K$-th round with user preference $W_K$, $\| \cdot \|_*$ stands for the trace norm of the user preference matrix $W_K$ (defined as the sum of the singular value of $W_K$) and $\gamma$ is the trace bound of $\|W_K\|_*$ to avoid the overfitting problem of user preference learning. Note that, the trace norm has been widely used as a convex surrogate of the nonconvex matrix rank function $\text{rank}(W_K)$ and achieves excellent empirical performance in [17], [51], [52]. We notice that when the dimension of content feature of the items $d$ equals to $m$ (i.e., let $x_j \in \{0, 1\}^d$ be the $j$-th user feature) Problem (2) can be reduced to the problem of matrix completion [4]. Note that, our online user preference learning method keeps all the previous user ratings. Currently, the storage of large scale data is not expensive while the bottleneck of online computation usually focuses on the efficiency of parameter estimation. We estimate the user preference $W_K$ from the previous parameter $W_{K-1}$ that minimizes the total loss function.

On the other hand, the benefits of the potential value of social relations have been well-recognized in social recommender systems, which could be applied to find the user’s like-minded neighbors and hence address the rating sparsity limitation [42], [7], [27]. We now consider that the social regularization is another regularization term for the optimization problem in Problem (2).

Consider the similarity matrix of users $S$ which is inferred from users’ social relations. We denote that $s_{ij}$ is the social similarity between the $i$-th user and the $j$-th user. Let $D$ be the diagonal matrix with $d_{ii} = \sum_j s_{ij}$, and $L = D - S$ be the Laplacian matrix. Based on the property of social relation, it is natural to assume that the socially similar users have similar preference in the rating matrix $R$. This assumption has been widely used in social recommendation and achieved excellent performance in [42], [7], [27]. Thus, the new regularization on the user preference matrix $W$ using the social similarity matrix $S$ can be achieved by minimizing [2]:
\[
\frac{1}{2} \sum_{k=1}^{n} \sum_{i,j=1}^{n} s_{ij} (f_{w_i}(x_k) - f_{w_j}(x_k))^2
\]
\[
= \frac{1}{2} \sum_{k} \left( \sum_{i,j} s_{ij} (x_k^T VW_i - x_k^T VW_j)^2 \right)
\]
\[
= \sum_{k} \left( \sum_{i} x_k^T VW_i \left( \sum_{j} s_{ij} W_j^T x_k \right) - \sum_{i,j} x_k^T W_i s_{ij} W_j^T x_k \right)
\]
\[
= \sum_{k} \left( \sum_{i} x_k^T VW_i \left( \sum_{j} s_{ij} W_j^T x_k \right) \right)
\]
\[
= \sum_{k} x_k^T (WDW^T - WSW^T)x_k
\]
\[
= tr \left( X^T W L W^T X \right). \tag{3}
\]

We then obtain the following optimization problem on the user preference matrix \( W_K \) at the \( K \)-th round, given by
\[
\min_{W_K} F_K(W_K) = \frac{1}{K} \sum_{k=1}^{K} tr \left( X^T W_K L W_K^T X \right)
\]
\[
+ \lambda tr \left( X^T W_K LW_K^T X \right)
\]
\[
s.t. \quad \|W_K\|_* \leq \gamma, \tag{4}
\]

where \( tr(\cdot) \) represents the graph regularization for the user preference matrix \( W_K \), and \( \lambda \geq 0 \) is a tradeoff parameter between the sequential loss function \( \frac{1}{K} \sum_{k=1}^{K} tr \left( X^T W_K LW_K^T X \right) \) and the graph regularization of user preference learning \( tr \left( X^T W_K LW_K^T X \right) \). Note that there is one fundamental difference between our formulation and the standard formulation of manifold regularization [2]. In our formulation \( W_K \) is the variable, whereas in the standard formulation \( X \) is the variable.

A common practice to solve Problem (4) is to use online ADMM [40], which decomposes the original problem into two smaller subproblems, and coordinates the solution of subproblems to compute the optimal solution. The main procedure of the ADMM algorithm to this problem can be described as follows:

1) Introduce the auxiliary variable for user preference matrix \( W \), and devise the augmented Lagrange function that decomposes the original problem into the problem of user preference learning and the problem of rank minimization with respect to the lagrange terms.
2) The problem of user preference learning can be solved with closed form solution.
3) The problem of rank minimization with respect to the lagrange terms can be solved using full singular value decomposition [55].
4) Iterates between Step 2 and Step 3 until the convergence of the algorithm.

However, we notice that the full singular value decomposition to the problem of rank minimization with respect to the lagrange terms is computationally expensive. Since the time complexity of full singular value decomposition is \( O(\min\{dn^2, d^3n\}) \) at each round, the cubic time complexity is very time consuming for online learning. In the experimental study, we observe that the running time of our online optimization method is only ten times the running time of one full singular value decomposition, which is much more efficient than the SVD-based online ADMM method. Instead of using full singular value decomposition, we introduce an efficient optimization method to solve the proposed problem in the next section.

5 The Online Optimization

In this section, we present a simple but efficient online optimization method to solve Problem (4). Given a sequential collection of user ratings at \( \Omega = \{\Omega_k, \ldots, \Omega_K\} \), the online optimization method aims to learn the user preference from \( W_1, \ldots, W_K \), where the trace norm ball of each \( W_k \) is bounded by \( \gamma \) (i.e., \( \|W_k\|_* \leq \gamma \)). That is, at each round \( k \), the optimization learns the user preference \( W_k \) based on the current user ratings at \( \Omega_k \) and the previous preference matrix \( W_{k-1} \).

Suppose the optimal loss value of bath preference learning is \( \min_{\|W\|_* \leq \gamma} F_K(W) \) and the user preference learned at the \( K \)-th round is \( W_K \), the goal of our online optimization method is to learn \( W_K \) such that the regret,
\[
Regret = F_K(W_K) - \min_{\|W\|_* \leq \gamma} F_K(W),
\]

is sublinear in \( K \).

We first have a brief discussion on the property of the trace norm constrained objective function in Problem (4). We observe that this objective function falls into the general category of convex constrained optimization, which can be solved by Frank Wolfe’s conditional gradient descent method [15]. Convex constrained optimization refers to the convex optimization problem with a convex objective function and a compact convex domain.

We notice that loss function of the proposed online optimization problem \( F_K(W_K) = \frac{1}{K} \sum_{k=1}^{K} tr(\Omega_k (X, R, \Omega_r) + \lambda tr(\Omega_k X^T W_K LW_K^T X)) \) is the convex objective function, and \( \|W\|_* \leq \gamma \) is the compact convex domain. A natural approach for solving Problem (4) is Frank Wolfe’s conditional gradient descent method which needs to evaluate the gradient of the objective function. Thus, in order to solve the convex constrained optimization in Problem (4), we need to evaluate the gradient of \( F_K(W_K) \) below.

Let \( e_i \in \{0,1\}^n \) be the \( i \)-th united vector. The gradient of the loss function \( F_K(W_K) \) is given by
\[
\nabla F_K(W_K) = \frac{1}{K} \sum_{r=1}^{K} \frac{1}{|\Omega_r|} \sum_{(i,j) \in \Omega_r, w_i = W_K(i)} 2(w_i^T x_j - r_{ij}) x_j e_i + 2\lambda XX^T W_K L,
\]
which can be verified in [33].
Therefore, we choose the variant of Frank Wolfe’s conditional gradient descent method to solve Problem (4) and the optimization algorithm (OGRPL-FW) for the proposed objective function in the following section.

5.1 The Optimization using OGRPL-FW

In this section, we introduce a conditional gradient method to solve the online graph regularized user preference learning problem using Frank Wolfe algorithm, denoted by OGRPL-FW.

The OGRPL-FW method can be decomposed into two procedures: direction-finding procedure and online updating procedure [15]. The direction-finding procedure is to find the Vₖ that minimizes ∇Fₖ(Wₖ)Vₖ where Vₖ is subject to ∥Vₖ∥₂ ≤ γ. Given the estimated Vₖ, the online updating procedure is given by Wₖ₊₁ = (1 − k⁻α)Wₖ + k⁻αVₖ where α ≥ 0 is the user specified parameter.

We now present the details of direction-finding procedure. We first consider that the constraint of the trace norm of W is the unit ball in the direction-finding procedure. Note that the use of unit ball constraint comes with no loss of generality, since the user preference matrix W can be re-scaled by an arbitrary constant [15]. Thus, subproblem in the direction-finding procedure amounts to approximating the top eigenvalue of the gradient of the proposed objective function [15], given by

$$V_k = \arg \min_{\|W\|_2 \leq \gamma} \{\nabla F_k(W_k) \cdot W\}$$

$$= \delta_1(\nabla F_k(W_k)),$$

where δ₁(∇Fₖ(Wₖ)) is the top eigenvalue of the gradient of the proposed objective function.

We then introduce an useful tool, that is, lanczos’ algorithm [20] for computing the top eigenvalue efficiently. Suppose that N(W) is the number of non-zeros in the matrix W. Lanczos’ algorithm takes the time complexity of $O(\frac{N(W)}{\sqrt{\varepsilon}})$ with ε-accurate low-rank solution. Therefore, the running time of the optimization over bounded trace-norm using the Lanczos’ algorithm is bounded as follows:

**Theorem 1:** ([20]) For any matrix $W \in \mathbb{R}^{m \times n}$, and $\varepsilon > 0$, Lanczos’ algorithm returns a pair of unit vectors (u, v) s.t. $u^T W v \geq \delta_1(W) - \varepsilon$, with high probability, using at most $O(\frac{N(W)}{\sqrt{\varepsilon}})$ arithmetic operations.

5.2 Algorithm and Analysis

The whole procedure of OGRPL-FW is summarized in Algorithm 1. Given a sequential collection of user ratings with indices $\Omega_1, \ldots, \Omega_K$, the method OGRPL-FW learns user preference $W_1, \ldots, W_K$ sequentially by solving the online constrained optimization method at each round. The computation of direction-finding procedure is in Line 3 and online updating procedure is in Line 4 of Algorithm 1. At each round k, the OGRPL-FW method computes the current user preference $W_k$ from user ratings at $\Omega_k$ and the preference estimated at the previous round $W_{k-1}$.

The main computation cost of OGRPL-FW in each iteration is the computation of the constrained optimization in Line 3, which has a linear cost of the nonzero entries in W (i.e., $O(N(W))$). Therefore, the total running time of Algorithm 1 is a linear cost of the nonzero entries in W. Suppose the matrix $W_*$ is the optimal solution to the batch optimization of user preference learning, the regret bound of Algorithm 1 for Problem (4) is $F_k(W_k) - F_k(W_*) \leq \frac{C}{\sqrt{T}}$, where C is a constant value, which is guaranteed by the property of the Frank-Wolfe method [13].

6 EXPERIMENTAL RESULTS

In this section, we conduct several experiments using the Douban dataset, to show the efficiency and effectiveness of our proposed approach GROPL-FW. The experiments are conducted by using Matlab, tested on machines with Linux OS Intel(R) Core(TM) Quad CPU 2.66Hz, and 32GB RAM.

6.1 Data Preparation

We collect the data from Douban¹, which is one of the largest online recommendation communities in China. Douban is a Chinese Web site providing user rating and recommendation services for movies and music, which was launched on March, 2005. Users can assign 5-scale integer ratings (from 1 to 5) to movies and music. It also provides Facebook-like social networking services, which allows users to find their friends through their email accounts.

We crawl the users’ social networks with their movie and music ratings from Douban, respectively. We obtain 3,186 friends through their email accounts.

¹. http://www.douban.com
TABLE 2
Statistics of User-Item Matrix of Douban and CIAO

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DoubanMovie</th>
<th>DoubanMusic</th>
<th>CIAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>User</td>
<td>Item</td>
<td>User</td>
</tr>
<tr>
<td>Min. Number of Ratings</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Max. Number of Ratings</td>
<td>9,215</td>
<td>1,790</td>
<td>6,159</td>
</tr>
<tr>
<td>Avg. Number of Ratings</td>
<td>129.1</td>
<td>27.8</td>
<td>137.1</td>
</tr>
</tbody>
</table>

![Fig. 2. Data Sparsity of DoubanMovie, DoubanMusic and CIAO Data](image1)

![Fig. 3. User Rating Distribution of DoubanMovie, DoubanMusic and CIAO Data](image2)

![Fig. 4. Item Rating Distribution of DoubanMovie, DoubanMusic and CIAO Data](image3)

then collect 7,649 users, 138,133 music, and 1,048,575 music ratings. The ratio of the observed rating to music rating matrix is 0.1%, and we consider the music rating matrix as sparse rating matrix. The statistics of the Douban user-item rating matrix and social friend network are summarized in Table 2 and Table 7, respectively. The data sparsity of both data is illustrated in Figures 2(a) and 2(b), respectively. The $y$-axis is the value of the number of users and $x$-axis denotes the number of user ratings. We can observe that the DoubanMusic data has a longer tail than the DoubanMovie data. Figures 3(a) and 3(b) illustrate the user rating distribution, and Figures 4(a) and 4(b) illustrate the item rating distribution over both datasets. The $y$-axis is the value of the number of users, $x$-axis in Figures 4(a) and 4(b) denotes the average item rating. We collect another dataset from CIAO\cite{2}, which is one of the largest product review sites. Users in CIAO can rate products with scores from 1 to 5 and they can also establish social relations (i.e., trust relations) with others. We crawl the data from CIAO and obtain 7,300 users, 1,000 products and 284,086 product ratings. The statistics of the CIAO user-item rating matrix and social trust network are summarized in Table 2 and Table 7, respectively. The data sparsity of the CIAO data is illustrated in Figure 2(c). The $y$-axis is the value of the number of users and $x$-axis denotes the number of user ratings. Figure 3(c) illustrates the user rating distribution, and Figure 4(c) illustrates the item rating distribution over the CIAO dataset. The $y$-axis is the value of the number of users.

TABLE 3
Results on MAE using DoubanMovie Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Online Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMTCF</td>
<td>0.7698 ± 2.27 × 10^-6</td>
<td>0.8203 ± 5.47 × 10^-3</td>
<td>0.9107 ± 3.03 × 10^-3</td>
<td>0.9966 ± 4.24 × 10^-6</td>
<td>1.0714 ± 4.07 × 10^-6</td>
</tr>
<tr>
<td>PA-MF</td>
<td>0.8444 ± 8.48 × 10^-6</td>
<td>0.9121 ± 7.21 × 10^-3</td>
<td>0.9588 ± 1.66 × 10^-4</td>
<td>0.9931 ± 2.46 × 10^-5</td>
<td>1.0175 ± 1.84 × 10^-5</td>
</tr>
<tr>
<td>OEMF</td>
<td>0.7312 ± 1.06 × 10^-5</td>
<td>0.7455 ± 2.89 × 10^-5</td>
<td>0.7532 ± 3.45 × 10^-4</td>
<td>0.7755 ± 1.86 × 10^-5</td>
<td>0.8677 ± 2.05 × 10^-5</td>
</tr>
<tr>
<td>MatchBox</td>
<td>0.6524 ± 2.45 × 10^-3</td>
<td>0.6548 ± 6.93 × 10^-4</td>
<td>0.6586 ± 1.65 × 10^-4</td>
<td>0.7084 ± 7.14 × 10^-4</td>
<td>0.8472 ± 1.20 × 10^-3</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>0.6094 ± 1.45 × 10^-6</td>
<td>0.6179 ± 6.05 × 10^-7</td>
<td>0.6303 ± 5.00 × 10^-9</td>
<td>0.6542 ± 1.80 × 10^-7</td>
<td>0.7264 ± 2.65 × 10^-6</td>
</tr>
</tbody>
</table>

TABLE 4
Results on RMSE using DoubanMovie Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Online Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMTCF</td>
<td>0.9902 ± 7.39 × 10^-6</td>
<td>1.0586 ± 2.06 × 10^-3</td>
<td>1.2916 ± 1.46 × 10^-3</td>
<td>1.5287 ± 1.60 × 10^-3</td>
<td>1.752 ± 4.90 × 10^-3</td>
</tr>
<tr>
<td>PA-MF</td>
<td>1.1299 ± 8.97 × 10^-5</td>
<td>1.3193 ± 3.63 × 10^-5</td>
<td>1.4541 ± 1.44 × 10^-3</td>
<td>1.5536 ± 2.49 × 10^-4</td>
<td>1.626 ± 1.19 × 10^-4</td>
</tr>
<tr>
<td>OEMF</td>
<td>0.8596 ± 4.05 × 10^-5</td>
<td>0.8979 ± 1.34 × 10^-4</td>
<td>0.9276 ± 4.20 × 10^-4</td>
<td>0.9968 ± 5.51 × 10^-5</td>
<td>1.263 ± 4.80 × 10^-5</td>
</tr>
<tr>
<td>MatchBox</td>
<td>0.7064 ± 1.91 × 10^-3</td>
<td>0.7089 ± 5.14 × 10^-4</td>
<td>0.7127 ± 1.86 × 10^-3</td>
<td>0.8237 ± 1.45 × 10^-3</td>
<td>1.0227 ± 1.31 × 10^-3</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>0.6185 ± 1.15 × 10^-5</td>
<td>0.6434 ± 6.05 × 10^-7</td>
<td>0.6739 ± 2.00 × 10^-8</td>
<td>0.7353 ± 1.25 × 10^-7</td>
<td>0.9247 ± 1.10 × 10^-6</td>
</tr>
</tbody>
</table>

TABLE 5
Results on MAE using DoubanMusic Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Online Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMTCF</td>
<td>0.8743 ± 6.95 × 10^-7</td>
<td>0.8797 ± 2.79 × 10^-6</td>
<td>0.9042 ± 1.29 × 10^-5</td>
<td>1.0058 ± 2.03 × 10^-4</td>
<td>1.0103 ± 1.66 × 10^-6</td>
</tr>
<tr>
<td>PA-MF</td>
<td>0.9514 ± 3.37 × 10^-5</td>
<td>1.0127 ± 1.64 × 10^-5</td>
<td>1.0641 ± 3.69 × 10^-6</td>
<td>1.1047 ± 2.60 × 10^-6</td>
<td>1.1278 ± 1.21 × 10^-5</td>
</tr>
<tr>
<td>OEMF</td>
<td>1.2777 ± 1.90 × 10^-4</td>
<td>1.3613 ± 3.53 × 10^-5</td>
<td>1.4884 ± 9.94 × 10^-5</td>
<td>1.7249 ± 1.10 × 10^-5</td>
<td>2.3102 ± 8.00 × 10^-8</td>
</tr>
<tr>
<td>MatchBox</td>
<td>0.6216 ± 5.05 × 10^-4</td>
<td>0.6215 ± 5.03 × 10^-4</td>
<td>0.6233 ± 6.54 × 10^-4</td>
<td>0.6294 ± 7.54 × 10^-4</td>
<td>0.8117 ± 6.20 × 10^-4</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>0.5277 ± 7.20 × 10^-7</td>
<td>0.5322 ± 2.45 × 10^-7</td>
<td>0.5373 ± 3.20 × 10^-7</td>
<td>0.5498 ± 1.13 × 10^-6</td>
<td>0.6044 ± 5.00 × 10^-7</td>
</tr>
</tbody>
</table>

TABLE 6
Results on RMSE using DoubanMusic Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Online Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMTCF</td>
<td>1.3075 ± 6.31 × 10^-6</td>
<td>1.3444 ± 2.05 × 10^-9</td>
<td>1.3597 ± 1.71 × 10^-6</td>
<td>1.8048 ± 1.59 × 10^-9</td>
<td>1.8068 ± 1.30 × 10^-9</td>
</tr>
<tr>
<td>PA-MF</td>
<td>1.4184 ± 6.41 × 10^-4</td>
<td>1.6566 ± 2.87 × 10^-4</td>
<td>1.8361 ± 4.88 × 10^-5</td>
<td>1.9841 ± 1.16 × 10^-4</td>
<td>2.0741 ± 9.43 × 10^-5</td>
</tr>
<tr>
<td>OEMF</td>
<td>0.8446 ± 1.86 × 10^-5</td>
<td>0.8701 ± 3.65 × 10^-6</td>
<td>0.9116 ± 1.68 × 10^-5</td>
<td>0.9965 ± 1.44 × 10^-6</td>
<td>1.1994 ± 2.00 × 10^-6</td>
</tr>
<tr>
<td>MatchBox</td>
<td>0.6582 ± 2.33 × 10^-4</td>
<td>0.6581 ± 3.22 × 10^-4</td>
<td>0.6608 ± 1.10 × 10^-3</td>
<td>0.6695 ± 5.33 × 10^-4</td>
<td>0.9728 ± 5.90 × 10^-4</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>0.5605 ± 9.80 × 10^-7</td>
<td>0.5732 ± 8.00 × 10^-8</td>
<td>0.5968 ± 2.21 × 10^-9</td>
<td>0.6508 ± 1.62 × 10^-6</td>
<td>0.8882 ± 4.05 × 10^-5</td>
</tr>
</tbody>
</table>

TABLE 7
Statistics of Douban and CIAO User Relation Matrix

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Douban</th>
<th>CIAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Number of Degrees</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Max. Number of Degrees</td>
<td>356</td>
<td>100</td>
</tr>
<tr>
<td>Avg. Number of Degrees</td>
<td>10.3</td>
<td>15.1</td>
</tr>
</tbody>
</table>

x-axis in Figure 3(c) denotes the average user rating while x-axis in Figure 4(c) denotes the average item rating.

6.2 Evaluation Criteria

We evaluate the quality of rating prediction by our proposed approach in comparison with other collaborative filtering methods using two widely used evaluation criteria, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). We randomly sample 90% of the observed ratings into training data and store the remaining 10% into testing data. So training and testing data do not have overlap. We denote the indices of testing data by $\Omega$. Using the notation above, the evaluation criteria MAE is defined as:

$$ MAE = \frac{1}{|\Omega_{test}|} \sum_{(i,j) \in \Omega_{test}} |r_{ij} - \hat{r}_{ij}|, $$

and the evaluation criteria RMSE is defined as:

$$ RMSE = \sqrt{\frac{1}{|\Omega_{test}|} \sum_{(i,j) \in \Omega_{test}} (r_{ij} - \hat{r}_{ij})^2}, $$

where $|\Omega_{test}|$ is the number of entries in the testing data.

From the definitions, we can see that a smaller MAE or RMSE value means a better performance. For each setting, we carry out the cross-validation on MAE and RMSE for ten times and record the mean and the standard deviation values.

6.3 Performance Evaluations and Comparisons

In this section, we compare our method with four online recommendation algorithms and two offline recommendation algorithms using the evaluation criteria MAE and RMSE below:
### TABLE 8

Results on MAE using CIAO Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR2</td>
<td>0.8162 ± 2.3×10^-3</td>
<td>0.811 ± 4.5×10^-3</td>
<td>0.847 ± 5.8×10^-3</td>
<td>0.948 ± 3.1×10^-3</td>
<td>1.092 ± 2.7×10^-3</td>
</tr>
<tr>
<td>HSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMTCF</td>
<td>0.8811 ± 3.3×10^-3</td>
<td>0.9147 ± 3.3×10^-3</td>
<td>0.9548 ± 3.3×10^-3</td>
<td>1.0411 ± 3.3×10^-3</td>
<td>1.3047 ± 2.3×10^-3</td>
</tr>
<tr>
<td>PAMF</td>
<td>1.1247 ± 3.6×10^-3</td>
<td>1.1874 ± 3.6×10^-3</td>
<td>1.2463 ± 3.6×10^-3</td>
<td>1.3628 ± 3.6×10^-3</td>
<td>1.5633 ± 3.6×10^-3</td>
</tr>
<tr>
<td>OEMF</td>
<td>0.9389 ± 2.4×10^-3</td>
<td>0.9961 ± 4.1×10^-3</td>
<td>1.0345 ± 2.3×10^-3</td>
<td>1.1088 ± 4.1×10^-3</td>
<td>1.3107 ± 3.2×10^-3</td>
</tr>
<tr>
<td>MatchBox</td>
<td>0.8297 ± 5.1×10^-3</td>
<td>0.8831 ± 5.7×10^-3</td>
<td>0.9197 ± 6.7×10^-3</td>
<td>0.9982 ± 5.4×10^-3</td>
<td>1.1929 ± 8.0×10^-3</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>1.0521 ± 8.4×10^-4</td>
<td>1.078 ± 8.7×10^-4</td>
<td>1.0993 ± 10^-3</td>
<td>1.1045 ± 5.7×10^-4</td>
<td>1.1862 ± 5.7×10^-4</td>
</tr>
</tbody>
</table>

### TABLE 9

Results on RMSE using CIAO Data: The best and comparable results are highlighted

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR2</td>
<td>1.0457 ± 1.3×10^-4</td>
<td>1.0513 ± 9.4×10^-4</td>
<td>1.0728 ± 9.5×10^-4</td>
<td>1.1045 ± 5.7×10^-4</td>
<td>1.1862 ± 5.7×10^-4</td>
</tr>
<tr>
<td>HSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMTCF</td>
<td>1.1136 ± 3.3×10^-3</td>
<td>1.1567 ± 3.6×10^-3</td>
<td>1.1977 ± 1.7×10^-3</td>
<td>1.2874 ± 3.4×10^-3</td>
<td>1.5561 ± 3×10^-3</td>
</tr>
<tr>
<td>PAMF</td>
<td>1.3607 ± 2.9×10^-3</td>
<td>1.4341 ± 2.6×10^-3</td>
<td>1.4991 ± 8.3×10^-3</td>
<td>1.6131 ± 1.2×10^-3</td>
<td>1.8068 ± 5.8×10^-3</td>
</tr>
<tr>
<td>OEMF</td>
<td>1.2147 ± 3.7×10^-3</td>
<td>1.2778 ± 6.4×10^-3</td>
<td>1.3153 ± 8×10^-3</td>
<td>1.3872 ± 1.0×10^-3</td>
<td>1.5899 ± 2.7×10^-3</td>
</tr>
<tr>
<td>MatchBox</td>
<td>1.0792 ± 5.7×10^-3</td>
<td>1.1396 ± 4.7×10^-3</td>
<td>1.1796 ± 2.5×10^-3</td>
<td>1.252 ± 1.6×10^-3</td>
<td>1.4507 ± 1.5×10^-3</td>
</tr>
<tr>
<td>OGRPL-FW</td>
<td>1.02 ± 4.1×10^-3</td>
<td>1.09 ± 3.7×10^-3</td>
<td>1.1444 ± 1.1×10^-3</td>
<td>1.1853 ± 7.3×10^-3</td>
<td>1.4197 ± 6.9×10^-3</td>
</tr>
</tbody>
</table>

Fig. 5. Online performance of our GROPL-FW method on MAE using DoubanMovie data

Fig. 6. Online performance of our GROPL-FW method on RMSE using DoubanMovie data

- **OMTCF**: the online multi-task collaborative filtering for online recommendation, which is solved by online gradient descent method with social regularization, described in [46];
- **PA-MF**: the online matrix factorization method under the online passive-aggressive learning framework [5] for collaborative filtering, described in [3];
- **OEMF**: the online evolutionary collaborative filtering algorithm, which tracks user interest over time in order to make better recommendation, described in [22];
- **MatchBox**: the online bayesian recommendation algorithm, which makes the use of content information of items in combination with collaborative filtering method, described in [39].
- **SR2**: the matrix factorization framework with social regularization for item recommendation, which incorporates social network information to improve recommender system, described in [28].
6.4 Online Performance Study

In this section, we study the online performance of our method with respect to the different percentage of training data (i.e., 90%, 50% and 10%). Figures 5(a), 5(b) and 5(c) show the online performance of our OGRPL-FW method on MAE, and Figures 6(a), 6(b) and 6(c) illustrate the experimental results on RMSE using DoubanMovie data. On the other hand, we illustrate the online performance of our method using DoubanMusic data on MAE in Figures 7(a), 7(b) and 7(c), and on RMSE in Figures 8(a), 8(b) and 8(c), respectively. The y-axis is the value of evaluation criteria MAE or RMSE and x-axis denotes the percentage of user rating samples with different percentage of training data. We observe that our method achieves good MAE and RMSE results after receiving more than 50% percentage of training samples on both datasets. On the other hand, our method can achieve better quality of rating prediction with more training data.

6.5 Efficiency Study

In this section, we study the efficiency of our method using two datasets. Figures 9(a) and 9(b) illustrate the running time of our method on DoubanMovie and DoubanMusic data, respectively. The y-axis is the running time (in sec) and x-axis denotes the percentage of user rating samples in the 90% of the training data. We notice that the trend of running time of our method is linear with respect to the percentage of samples. The total running time of our online learning method is less than 300 seconds on DoubanMovie data and less than 1,600 seconds on DoubanMusic data. However, the running time of only one full singular value decomposition is 47 seconds on DoubanMovie data and 682 seconds on DoubanMusic data. Compared with the batch matrix completion and factorization

- **HSR**: the matrix factorization framework, which explores the implicit hierarchical structures for recommender systems, described in [47].

Tables 3 and 4 show the evaluation results of all the algorithms on MAE and RMSE using DoubanMovie data, Tables 5 and 6 illustrate the evaluation results using DoubanMusic data. We also evaluate the performance of our method on MAE and RMSE using CIAO data in Tables 8 and 9, compared with both online and offline algorithms. For both datasets, we vary the percentage of training data (i.e., 90%, 70%, 50%, 30% and 10%) to evaluate the performance of all the algorithms. For each experimental setting, we carry out the cross-validation ten times and report the mean and the standard deviation values.

We can observe that our OGRPL-FW method achieves the best performances, owning to two main reasons. First, the formulation of online matrix factorization is non-convex optimization problem while the formulation of our online user preference learning is a convex optimization. Thus, the performance of our proposed online gradient descent method to the optimization problem is better than those gradient descent methods to the non-convex problem. Second, the loss functions of both the proposed online collaborative filtering algorithms and our method are in the category of data reconstruction, where the training data can be reconstructed by the learned model. Since the storage of large scale data is not expensive currently while the bottleneck of online computation usually focuses on the efficiency of parameter estimation. Our method keeps the historical user rating data and learn the current user preference from the user preference of the last round and the collected user ratings.
methods based on singular value decomposition, our online method is much efficient.

6.6 Parameter Study

In our approach, there is one essential parameter, that is, the graph regularization parameter $\lambda$. When the value of $\lambda$ becomes small, the problem of online social recommendation can be reduced to online user preference learning, which is based on the user-item rating matrix only. We vary the parameter $\lambda$ to investigate the effect of graph regularization for online social recommendation using 50% of the training data. We vary the value of parameter $\lambda$ of our method from $10^{-7}$ to $10^{-2}$ on DoubanMovie data and from $10^{-8}$ to $10^{-3}$ on DoubanMusic data. As we can see, the performance of our OGRPL-FW method is very stable with respect to the parameter $\lambda$, which achieves consistently good performance.

7 Conclusion

We presented a new framework of online social recommendation from the viewpoint of online user preference learning, which incorporates both collaborative user-item relationship as well as item content features into an unified preference learning process. We consider that the user model is the preference function which can be online learned from the user-item rating matrix. Furthermore, our approach integrates both online user preference learning and users’ social relations seamlessly into a common framework for the problem of
online social recommendation. In this way, our method can further improve the quality of online rating prediction for the missing values in the user-item rating matrix. We devise an efficient iterative procedure, OGRPL-FW to solve the online optimization problem. We conduct extensive experiments on several large-scale datasets, in which the encouraging results demonstrate that our proposed algorithm achieves better performance than the state-of-the-art online recommendation methods. In the future, we will explore the non-linear user preference learning function as the user model for the problem of online social recommendation.

REFERENCES


Zhou Zhao received the BS and the Ph.D. degrees in computer science from the Hong Kong University of Science and Technology (HKUST), in 2010 and 2015, respectively. He is currently a lecturer with the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.

Yueling Zhuang received his B.Sc., M.Sc. and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1986, 1989 and 1998 respectively. From February 1997 to August 1998, Yueling Zhuang was a visiting scholar at Prof. Thomas Huang’s group, University of Illinois at Urbana-Champaign. Currently, He is a full professor and the Dean of the College of Computer Science, Zhejiang University. His research interests mainly include artificial intelligence, multimedia retrieval, computer animation and digital library.

Xiaofei He received the B.S. degree in computer science from Zhejiang University, Hangzhou, China, in 2000, and the Ph.D. degree in computer science from the University of Chicago, Chicago, IL, USA, in 2005. He is currently a Professor with the College of Computer Science, Zhejiang University. Prior to joining Zhejiang University, he was a Research Scientist with Yahoo! Research Labs, Burbank, CA, USA.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining and information retrieval.