Learning Discriminative Recommendation Systems with Side Information

Feipeng Zhao
Computer and Information Sciences
Temple University, Philadelphia, USA
feipeng.zhao@temple.edu

Yuhong Guo
School of Computer Science
Carleton University, Ottawa, Canada
yuhong.guo@carleton.ca

Abstract

Top-N recommendation systems are useful in many real world applications such as E-commerce platforms. Most previous methods produce top-N recommendations based on the observed user purchase or recommendation activities. Recently, it has been noticed that side information that describes the items can be produced from auxiliary sources and help to improve the performance of top-N recommendation systems; e.g., side information of the items can be collected from the item reviews. In this paper, we propose a joint discriminative prediction model that exploits both the partially observed user-item recommendation matrix and the item-based side information to build top-N recommendation systems. This joint model aggregates observed user-item recommendation activities to produce the missing user-item recommendation scores while simultaneously training a linear regression model to predict the user-item recommendation scores from auxiliary item features. We evaluate the proposed approach on a number of recommendation datasets. The experimental results show that the proposed joint model is very effective for producing top-N recommendation systems.

1 Introduction

Top-N recommendation systems automatically predict the missing recommendation scores over the set of product items for each consumer, and recommend a short list of N items with highest scores to each consumer. With the increasing popularity of online shopping, effective top-N commercial recommendation systems become of great importance in helping consumers to find their interested items and hence encouraging online item purchases. For example, a top-N recommendation system can effectively help consumers to find their potential interested products from Amazon.com and buy them online, and help users to find movies that best match their interests on Netflix. In such real world application domains, effective top-N recommendation systems can make significant commercial impacts.

Many algorithms have been developed in the literature to build top-N recommendation systems [Ricci *et al.*, 2011]. Most of them use a user-item rating matrix to build the recommendation models. A classical technique is collaborative filtering (CF) [Schafer *et al.*, 2007], which models the relationships between users and the correlations between items to identify new user-items relationship scores. Ranking methods have also demonstrated good performance for top-N recommendations [Weimer *et al.*, 2008; Steck, 2010; Chen and Pan, 2013; Aiolli, 2014; Park *et al.*, 2015]. In addition to CF and ranking, sparse aggregation methods that exploit linear correlations between items have been explored in a few works to improve top-N recommendation performance [Ning and Karypis, 2011; Cheng *et al.*, 2014; Kabbur *et al.*, 2013; Christakopoulou and Karypis, 2014]. Nevertheless, all these methods focus solely on the user-item rating (or purchase) matrix, without exploring any additional information.

In many applications, in addition to the user-item rating or purchase matrix, item-based side information such as product reviews, book reviews, item comments, and movie plots can be easily collected from the Internet. This abundant item-based information can be used for recommendation systems. The work in [Mooney and Roy, 2000] used book titles, reviews, and comments as item features; the work in [Melville *et al.*, 2002] used item contents to enrich the sparse rating matrix. These recommendation systems however either only focus on user profiles or item contents without fully considering the user-item rating matrix or simply use item contents to preprocess the sparse rating matrix to produce small improvements. Moreover, they have not addressed the top-N recommendation problems. A recent work [Ning and Karypis, 2012] incorporated item-based side information to improve top-N recommender systems, which however assumes that user-item matrix and item-based side information matrix are reproduced by the same linear aggregation model and does not exploit the discriminative power of the side information.

In this paper, we propose a joint discriminative prediction model that exploits both the partially observed user-item recommendation (rating/purchase) matrix and the item-based side information such as item contents (product reviews and movie plots) to build top-N recommendation systems. The item information used in this work can be viewed as descriptions of the items and hence can produce feature representation vectors for the items. We propose a novel component that predicts the user-item recommendation (purchase) scores

discriminatively from the item features with linear regression models, and integrate this component together with an aggregation model on the user-item matrix to jointly predict the unknown user-item recommendation scores. The joint prediction of the recommendation scores from both information sources is expected to increase the performance of top-N recommendation systems. We formulate this joint prediction model as a convex minimization problem, which can be solved using a projected gradient descent algorithm. We evaluate the proposed approach by conducting experiments on several real world datasets. The experimental results show that the proposed approach can outperform both popular top-N recommendation methods that only use user-item matrix and state-of-the-art algorithms that exploit additional side information. *Moreover*, our empirical study also reveals an interesting observation: the discriminative prediction component that predicts the recommendation scores from the item features (i.e., side information) contributes to most of the recommendation capacity of our joint prediction model, and the prediction component itself already outperforms other comparison methods. This observation encourages researchers to explore prediction models for building top-N recommendation systems.

2 Related Work

In this section, we review related works, including standard top-N recommendation systems and recent recommendation approaches that exploit item-based side information.

Collaborative filtering (CF) technology is widely used in top-N recommendation systems. CF methods, which explore the purchase history of all the users, relationships between the users and correlations between the items, can be divided into two types, neighborhood-based CF and model-based CF. Neighborhood-based CF methods exploit similarities between all the items and users to make recommendations [Deshpande and Karypis, 2004; Verstrepen and Goethals, 2014]. Model-based CF methods learn latent factors for users and items to reconstruct the user-item matrix. For example, the work in [Cremonesi *et al.*, 2010] developed a singular value decomposition (SVD) method to directly learn latent user factors and item factors. Though simple, this method has good performance for top-N recommendations. In [Hu *et al.*, 2008; Sindhwani *et al.*, 2010], a weighted regularized matrix factorization (WRMF) model is formulated to learn the latent factors based on implicit user-item feedbacks. The work in [Shi *et al.*, 2012b] proposed to learn the latent factors by directly maximizing the mean reciprocal rank. The work in [Liu and Aberer, 2014] addressed top-N recommendation in dynamic situations.

Weimer et.al [Weimer *et al.*, 2008] considered the top-N recommendation problem as a ranking problem, and used maximum margin matrix factorization to optimize ranking scores instead of rating scores. Following this work, a number of methods have been developed in this direction. The authors of [Rendle *et al.*, 2009] proposed a Bayesian personalized ranking optimization method for item recommendation with matrix factorization. The work in [Steck, 2010] addressed top-N item recommendation in the non-random missing situation by optimizing the metric of area under top-N curve. The work in [Chen and Pan, 2013] assumed that a user's preference is on a set of products instead of only one, and proposed to learn pairwise preferences over item-sets. The work in [Aiolli, 2014] focused on implicit feedbacks where preferences are given in the form of binary ratings, and proposed to optimize the ranking score within a margin maximization paradigm. The authors of [Park *et al.*, 2015] developed a large-scale optimization algorithm for ranking based matrix completion.

In addition to collaborative filtering and ranking methods, a few sparse linear aggregation methods have recently been developed in the literature. The work in [Ning and Karypis, 2011] proposed a novel sparse linear method (SLIM) to perform top-N recommendation. It learns a sparse aggregation coefficient matrix for items by solving an ℓ_1-norm and ℓ_2-norm regularized optimization problem, and aggregates the user purchase/rating profiles to produce the top-N recommendations. The work in [Kabbur *et al.*, 2013] developed a factored item similarity model to tackle the situation where user-item matrix is very sparse and the SLIM method may fail to capture the item-item similarity. It learns the similarity matrix as the product of two low dimensional latent factor matrices and overcomes the sparsity problem. The work in [Cheng *et al.*, 2014] proposed a low rank SLIM method to improve SLIM, which adds a nuclear norm constraint to enforce a low rank aggregation matrix. The methods in [Christakopoulou and Karypis, 2014] extend item pairwise similarities to higher orders, and capture more information with high order item correlations to better reconstruct the user-item matrix. In [Sedhain *et al.*, 2016], the authors exploit user-user similarities instead of item-item similarities to produce recommender systems based on linear logistic regression.

Instead of only focusing on the user-item matrix, a few works in the literature have incorporated contextual information [Zheng *et al.*, 2014; Shi *et al.*, 2012a], user social networks [Yang *et al.*, 2012], and information from linked open data [Ostuni *et al.*, 2013] to improve recommendation performance. Moreover, a number of methods have used more easily obtained item-based information to design recommendation systems. For example, the work in [Mooney and Roy, 2000] used a Bayesian learning algorithm to build a content-based recommender system based on the item-based features, without using the user-item recommendation matrix. The work in [Melville *et al.*, 2002] used item contents to pre-process the sparse rating matrix to overcome the sparsity of user-item purchase matrix. These two methods though are not designed for top-N recommendations, they verified the usefulness of item contents for item recommendation systems. More recently, the authors of [Ning and Karypis, 2012] proposed a collective SLIM (cSLIM) method to incorporate item-based side information such as the reviews of the items into the SLIM to improve top-N recommender systems. This method assumes that user-item purchase matrix and item-based side information matrix are reproduced by the same sparse linear aggregation matrix, which is a very strong assumption in many real applications. The work in [Zhao *et al.*, 2016] incorporated side informaiton into low-rank collaborative filtering. A few recent works [Bao *et al.*, 2014;

McAuley and Leskovec, 2013; Xu *et al.*, 2014] have also exploited user reviews for building recommender systems. However these methods require the reviews to be directly associated with the user ratings, while our approach can exploit reviews from auxiliary resources that have no direct relationships with the target users.

3 Proposed Approach

In this section, we present a novel joint prediction model to produce personalized top-N recommendation systems. We assume an item purchase matrix $Y = \{0,1\}^{n\times t}$ over n product items and t users is given, where $Y_{ij} = 1$ indicates an observed purchase relationship between the i-th product item and the j-th user while $Y_{ij} = 0$ indicates an unknown relationship between them. Note that an unobserved entry does not mean that the corresponding user is not interested in the particular product item. It only shows that the purchase record is not observed in the given user's past purchase history. The recommendation system will predict the unobserved entries and recommend to each user the top-N items with highest prediction scores from the unobserved item pool. We also assume the existence of item-based side information, which could be in the form of product reviews for e-commerce applications or film plots for movie recommendations. Different from some literature works [Bao *et al.*, 2014; Xu *et al.*, 2014] that exploited product reviews to build direct relationships between the considered users and products, we only require reviews provided as "side-information", which have no assumed connections to the target group of users. For simplicity, we assume the side-information will be presented as an item-feature matrix $X \in \mathbb{R}^{n\times d}$, whose each row contains features for a product item. We will use both X and Y to jointly recover an item-user recommendation matrix $\widehat{Y}$, which contains the predicted recommendation scores for all product items. We then can produce the top-N most interesting items for each target user based on the predicted scores.

In the rest of the paper, we will use $\mathbf{1}$ to denote a column vector of all 1s, assuming the size of the column vector can be inferred from the context. We use I to denote an identity matrix; use $\|X\|_F$ to denote the Frobenius norm of matrix X; use $X_{:j}$ to denote the j-th column and X_{ij} to denote the (i,j)-th entry of matrix X respectively.

3.1 A Joint Prediction Model

We aim to predict the unobserved recommendation entries in the item-user matrix by exploiting both the given item-user matrix and the item-based side information matrix. The proposed joint recommendation prediction model hence has two components, a self-recovery component that recovers a full item-user recommendation matrix $\widehat{Y}$ from Y and a linear regression prediction component that predicts $\widehat{Y}$ from the item-based side information matrix X. We will first describe each component below and then present the joint model.

Self-Recovery of the Recommendation Matrix

Given the partially observed item-user recommendation matrix Y, we propose to automatically recover the full recommendation matrix $\widehat{Y}$ and fill the recommendation scores in the unobserved entries of Y by aggregating the existing item-user recommendations. Specifically, we assume an unobserved recommendation score for the j-th user on the i-th item, Y_{ij}, can be calculated as a nonnegative linear aggregation of the observed item recommendations for the j-th user; that is

$$\widehat{Y}_{ij} = \mathbf{w}^T Y_{:j}$$

for $\mathbf{w} \geq 0$. Such a linear recovery model can be built for any items, $\widehat{Y}_{:j} = WY_{:j}$. The principal behind the linear recovery models is that one assumes the items chosen by the same user have nonnegative linear correlation patterns. Assuming the same linear aggregation pattern for each item is shared across different users, we then have a matrix self-recovery formulation $\widehat{Y} = WY$ with $W \geq 0$, where W is an $n \times n$ linear aggregation coefficient matrix that is shared across all the t users. The matrix self-recovery aims to recover the unobserved recommendation entries without affecting much the observed recommendation scores. Hence, to pursue a meaningful matrix self-recovery we should minimize the changes of the recovered matrix $\widehat{Y}$ from the pre-given item-user recommendation matrix Y. Moreover, to avoid the trivial solution of setting W as an identity matrix, we enforce the diagonal of W to be zeros. This leads to the following matrix self-recovery problem:

$$\min_{W} \quad \|Y - WY\|_F^2 \quad \text{s.t.} \quad W \geq 0, \ \ \text{diag}(W) = 0. \qquad (1)$$

The constraints force each entry to be recovered from the other entries for the same user and hence encourage the statistical discovery of consistent item aggregation models across all the users. Moreover, with this constrained learning problem, the observed entries of the recommendation matrix Y may not be exactly maintained in the recovered matrix WY. To overcome this problem, we simply combine the original observed entries and the recovered unobserved entries together to form our final recovered item-user recommendation matrix $\widehat{Y}$; that is, we set $\widehat{Y} = Y + (WY) \circ (1 - Y)$, where $\circ$ denotes the Hadamard product operator.

Discriminative Prediction from the Side Information

Given the item-based side information expressed as an item-feature matrix X, where each feature vector describes the properties of a given product item, we propose to learn a prediction model from the item-feature matrix X to predict the user ratings. Our intuition is that though the item-based side information is common to all the users, different parts of the item properties may correspond to the interests of different users. That is, different users may have different tastes, reflected by the recommendation scores or purchase activities, over the same item, which can be related to the different features or feature subsets of the item. We hence assume the recommendation scores/purchase activities of each user over the set of product items can depend on the properties of items, and propose to predict the recommendation scores of each user over a product item from the feature vector of this product item. In particular, we can use a linear regression model, $f_j(\mathbf{x}) = \mathbf{x}^T\mathbf{q} + b$, for the j-th user to predict his recommendation score y for the item described with feature vector

$\mathbf{x}$. Given the observed item-feature matrix X and the recovered item-user recommendation matrix $\widehat{Y}$, the linear regression model f_j for the j-th user can be trained by minimizing a least squares regression loss $||X\mathbf{q} + b\mathbf{1} - \widehat{Y}_{:,j}||^2$ over the model parameters $\mathbf{q}$ and b. For all the t users, we will then train t linear regression models by minimizing the following regularized least squares loss function

$$\min_{Q,b} \quad \|XQ + \mathbf{1}\mathbf{b}^T - \widehat{Y}\|_F^2 + \beta\|Q\|_F^2 \tag{2}$$

where $Q \in R^{d\times t}$ and $\mathbf{b} \in R^t$ are the parameters for the t linear regression models, the regularization term over Q is used to avoid over-fitting. When $\widehat{Y}$ is unknown, these linear regression models will contribute to the prediction of the unknown entries of $\widehat{Y}$. This prediction component is a novel component, which is fundamentally different from the previous methods in the literature by predicting the recommendation scores directly from the item-based side information.

Integration Model

With the bridge equation $\widehat{Y} = Y + (WY) \circ (1-Y)$, finally we can integrate the two components presented above, the recommendation matrix self-recovery component (i.e., Eq.(1)) and the linear regression prediction component (i.e., Eq.(2)), into the following joint matrix prediction model:

$$\begin{aligned}\min_{W,Q,b} \quad & \left\|XQ+\mathbf{1}\mathbf{b}^T-(Y+(WY)\circ(1-Y))\right\|_F^2 \\ & + \beta\|Q\|_F^2 + \gamma\|Y-WY\|_F^2 \\ \text{s.t.} \quad & W \geq 0, \quad \text{diag}(W) = 0\end{aligned} \tag{3}$$

where the trade-off parameter γ is introduced to balance the contribution of the two components. This joint model integrates information from both the partially observed item-user recommendation/purchase matrix Y and the item-feature matrix X, and is expected to further improve top-N recommendation systems.

In terms of learning a linear aggregation model to reconstruct the item-user matrix. our proposed method is related to the SLIM method and the cSLIM method developed in the literature. However, SLIM does not exploit side information, while cSLIM exploits the item-based side information with the same aggregation matrix W used for the matrix self-recovery. Our proposed approach is significantly different from these two methods in exploiting the item-based side information with linear regression models and integrating the two components via joint predictions.

3.2 Optimization Algorithm

The learning problem we formulated in Eq.(3) is a joint *convex* minimization problem over two sets of parameters, the linear regression model parameters $\{Q, \mathbf{b}\}$ and the linear aggregation coefficient matrix W. For a fixed coefficient matrix W, the minimization problem in Eq.(3) over Q and $\mathbf{b}$ becomes a standard linear regression problem, which has the following closed-form solution:

$$\mathbf{b} = \frac{1}{n}\left(Y + (WY)\circ(1-Y) - XQ\right)^\top \mathbf{1} \tag{4}$$

$$Q = (X^\top HX + \beta I)^{-1}X^\top H(Y + (WY)\circ(1-Y)) \tag{5}$$

Algorithm 1 Projected Gradient Descent Algorithm

Input: X, Y, parameters $\beta > 0, \gamma > 0$.
Initialize W as zeros.
while not converged **do**
1. find an optimal step size $\tau^* \in [0, 1]$ with line search
2. gradient descent: $W = W - \tau^*\nabla f(W)$
3. project onto feasible set:
$W = \max(W, 0), \quad \text{diag}(W) = 0$
end while

where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is a centering matrix. By plugging these solutions back into the objective function of Eq.(3), we can reformulate Eq.(3) equivalently into the following minimization problem over W:

$$\begin{aligned}\min_{W} \quad & \|B(Y + (WY)\circ(1-Y))\|_F^2 + \\ & \beta\|A(Y + (WY)\circ(1-Y))\|_F^2 + \gamma||Y - WY||_F^2 \\ \text{s.t.} \quad & W \geq 0, \quad \text{diag}(W) = 0\end{aligned} \tag{6}$$

where $A = (X^THX + \beta I)^{-1}X^TH$ and $B = H(XA - I)$. This remains to be a linear constrained *convex* minimization problem. We next develop a projected gradient descent algorithm to solve it.

Let $f(W)$ denote the objective function of Eq.(6). The projected gradient descent algorithm will iteratively minimize $f(W)$ subjecting to the constraints. In each iteration, given the current W, the gradient of the objective function can be computed as:

$$\begin{aligned}\nabla f(W) = & 2\gamma(WY - Y)T^T + 2((1-Y)\circ \\ & ((B^TB + \beta A^TA)(Y + (WY)\circ(1-Y))))Y^T\end{aligned} \tag{7}$$

With this gradient, it will first take a gradient descent step over W and then project the new W into the feasible set defined by the constraints. The overall algorithm is described in Algorithm 1, where the step size parameter τ for the gradient descent is determined using a standard backtracking line search:

$$\tau^* = \underset{0\leq\tau\leq 1}{\text{argmin}} \quad f(W - \tau\nabla f(W))$$

4 Experiment

We conducted experiments on a few real world datasets. In this section, we will first describe the experimental setup and then present experimental results and discussions.

4.1 Experimental Setup

We used the real world Amazon user rating and product review data for five categories of products, *Beauty*, *Office*, *Sports& Outdoors*, *Health* and *Gourmet Foods*, to conduct experiments. For each category, the original dataset we downloaded contains the ratings of the users over the products and the product reviews. Following [Ning and Karypis, 2011; 2012], we converted the rating values to 1s and produced implicit feedback matrices to use. The initial implicit feedback matrices are extremely sparse. We selected the top ten

thousand items based on the number of purchases for each item, and then selected the top four thousands of users to use. We further preprocessed the datasets by selecting users with at least 3 purchases and filtering the items without any purchases. Finally we obtained five item-user matrices.

For each item-user purchase (feedback) matrix produced, we used the product reviews as the item-based side information. The reviews are given in plain text and we preprocessed each item review in the following way. We first extracted unigram features from the review articles, and then removed the stop words and selected the top 5000 frequent unigram features as the item features. Finally each product item is represented as a bag-of-word feature vector in terms of these 5000 unigram features. We used the term-frequency feature values as the item-feature data.

Comparison Methods We compared our proposed method with the following methods for top-N recommendation systems: (1) Pure Singular Value Decomposition (pureSVD) [Cremonesi *et al.*, 2010]. This method performs singular value decomposition directly on the item-user purchase matrix, and reconstructs the matrix with the top subsets of singular vectors. (2) Weighted Regularized Matrix Factorization (WRMF) [Hu *et al.*, 2008]. WRMF extracts the latent factors for the users and items by performing weighted regularized matrix factorization. The reconstructed item-user matrix based on the extracted factors is then used for top-N recommendations. (3) Sparse Linear Method (SLIM) [Ning and Karypis, 2011]. SLIM learns an item-item aggregation sparse coefficient matrix by minimizing a constrained reconstruction loss. It uses the coefficient matrix to reconstruct the item-user purchase matrix. (4) Collective SLIM (cSLIM) [Ning and Karypis, 2012]. cSLIM incorporates side information into the SLIM model by enforcing the item-feature matrix to share the same aggregation matrix with the item-user matrix. (5) Inductive Matrix Completion (IMC) [Jain and Dhillon, 2013]. IMC is a state-of-art matrix completion method that exploits side information. We applied it to produce top-N recommendation systems.

Evaluation Metrics We evaluated our proposed method and the comparison methods using 5-fold Leave-One-Out-Cross-Validation. For each fold, the dataset is divided into a training set and a testing set: We randomly chose one transaction for each user and placed it to the test set, and the rest is used as training set. The training set is used to perform training. Then a ranked list of top N items are generated from the unobserved items for each user according to their scores in the reconstructed item-user matrix. The test results are produced by comparing the selected top N items for each user to his observed test set item. If there is a match between the test set item and the top N items, it is counted as one hit. The default N value used in our experiments is 10. We used two measurements to evaluate the test results: Hit Rate(HR) and the Average Reciprocal Hit Rate (ARHR) [Deshpande and Karypis, 2004], which are defined as

$$HR = \frac{\#hits}{\#users}, \qquad ARHR = \frac{1}{\#users}\sum_{i=1}^{\#hits}\frac{1}{p_i} \quad (8)$$

Table 1: Comparison results of top-N Recommendations on the five datasets. Bold font indicates the best results.

Method	Beauty				
	Params			HR(%)	ARHR(%)
PureSVD	100	-	-	33.1 ± 0.3	29.3 ± 0.2
WRMF	10	50	-	36.8 ± 0.3	31.9 ± 0.2
SLIM	10	0.01	-	39.0 ± 0.2	35.2 ± 0.3
cSLIM	10	0.01	1e-6	39.4 ± 0.3	35.2 ± 0.3
IMC	200	1	-	39.3 ± 0.2	35.4 ± 0.2
Proposed	200	0.1	-	$\mathbf{44.8 \pm 0.4}$	$\mathbf{39.7 \pm 0.3}$
Method	**Office**				
	Params			HR(%)	ARHR(%)
PureSVD	100	-	-	11.8 ± 0.3	9.1 ± 0.2
WRMF	10	50	-	18.3 ± 0.3	11.8 ± 0.2
SLIM	10	0.01	-	20.1 ± 0.4	13.7 ± 0.3
cSLIM	10	0.01	1e-6	21.2 ± 0.4	14.2 ± 0.3
IMC	200	0.001	-	20.9 ± 0.3	14.3 ± 0.2
Proposed	200	0.1	-	$\mathbf{27.6 \pm 0.5}$	$\mathbf{18.2 \pm 0.4}$
Method	**Sports&Outdoors**				
	Params			HR(%)	ARHR(%)
PureSVD	200	-	-	38.6 ± 0.5	34.8 ± 0.5
WRMF	10	50	-	41.2 ± 0.3	36.5 ± 0.4
SLIM	1	0.01	-	42.4 ± 0.4	39.0 ± 0.3
cSLIM	1	0.001	1e-7	42.7 ± 0.4	39.1 ± 0.3
IMC	200	1	-	45.0 ± 0.4	40.6 ± 0.3
Proposed	200	0.05		$\mathbf{46.9 \pm 0.4}$	$\mathbf{42.3 \pm 0.4}$
Method	**Gourmet Foods**				
	Params			HR(%)	ARHR(%)
PureSVD	100	-	-	9.9 ± 0.3	5.6 ± 0.2
WRMF	10	50	-	14.0 ± 0.2	7.9 ± 0.2
SLIM	10	0.01	-	14.3 ± 0.1	9.1 ± 0.1
cSLIM	10	0.01	1e-6	15.3 ± 0.2	9.3 ± 0.1
IMC	200	0.001	-	14.3 ± 0.2	8.4 ± 0.1
Proposed	200	0.05	-	$\mathbf{23.9 \pm 0.1}$	$\mathbf{13.7 \pm 0.1}$
Method	**Health**				
	Params			HR(%)	ARHR(%)
PureSVD	100	-	-	21.6 ± 0.3	18.2 ± 0.3
WRMF	10	50	-	26.2 ± 0.3	21.5 ± 0.3
SLIM	10	0.01	-	27.0 ± 0.4	23.8 ± 0.3
cSLIM	1	0.01	1e-6	28.7 ± 0.4	24.0 ± 0.3
IMC	200	0.001	-	28.1 ± 0.2	24.3 ± 0.2
Proposed	200	0.1	-	$\mathbf{32.2 \pm 0.4}$	$\mathbf{27.5 \pm 0.3}$

The *params* columns contain the parameter setting for each approach. *PureSVD* has one parameter f, indicating the number of latent factors. *WRMF* has two parameters, the regularization parameter λ and the latent factor dimension f. *SLIM* has two parameters, the ℓ_2 and ℓ_1 norm regularization parameters β and λ. Beyond β and λ, *cSLIM* has an additional side information weight parameter α. *IMC* has two parameters, hidden dimension f and regularization parameter λ. The proposed method has two trade-off parameters, β and γ.

where #users is the total number of users, #hits is the number of hits in the top-N recommendations across all users, and p_i is the position of the test item in the ranked recommendation list for the i-th hit. ARHR is a weighted version of HR, which takes the ranking position of the test item in the top-N recommendation list into account.

4.2 Comparison Results

We compared the five comparison methods (*pureSVD, WRMF, SLIM*, *cSLIM* and *IMC*) with the proposed approach on the five Amazon datasets using the evaluation metrics HR and ARHR. The average results and standard deviations for all the methods are reported in Table 1. We can see that within the three methods that do not exploit side information, *SLIM* outperforms the other two methods, *pureSVD* and *WRMF*. By using the item-based side information, *cSLIM* consistently outperforms all the three methods that do not exploit side information across all the datasets, and *IMC* outperforms the three methods on four out of the five datasets. This suggests that side information is useful, which is consistent with the results reported in [Ning and Karypis, 2012]. However, the improvements produced by *cSLIM* and *IMC* over the other three methods, especially over *SLIM*, are relatively small. Our proposed approach on the other hand consistently outperforms all the five comparison methods across all the datasets. The improvements achieved by our approach are notably large. For example, on *Office*, the proposed approach outperforms *cSLIM* by 6.4% and 4.0% in terms of HR and ARHR respectively, and outperforms *IMC* by 6.7% and 3.9% in terms of HR and ARHR respectively. Theses results demonstrate that our proposed approach provides an effective mechanism for exploiting the item-based side information to improve top-N recommendation systems.

4.3 Component-Wise Study

The proposed approach integrates two components, the self-recovery component and the linear regression prediction component, together to jointly perform item-user matrix reconstruction. How does each component contribute to the final top-N recommendation? Which component is more important? To answer these questions, we conducted another set of experiments to compare the proposed approach with its two individual components. The linear prediction component based on the side information can be obtained by simply setting $\gamma = 0$ in the proposed objective function Eq.(3) to drop the self-recovery component. The self-recovery component on item-user purchase matrix can be obtained by dropping the linear regression models from Eq.(3). This component can be viewed as a variant of the SLIM method without the regularization terms on W.

We compared the two components with the proposed joint model on all the five datasets. The comparison results are reported in Table 2. We can see that the performance of the prediction component with side information outperforms the self-recovery component with only purchase matrix with large margins across all the datasets. By comparing the results in both Table 1 and Table 2, we can see that the *linear prediction* component even consistently outperforms the most effective comparison methods, *cSLIM* and *IMC*, across all the datasets. This suggests that our novel prediction component that uses linear regression models to predict the item-user recommendation scores from the item-based side information is very effective. Nevertheless, our proposed approach that integrates the two components together consistently outperforms each individual component, which suggests the two components contain complementary information and our proposed model can effectively capture such information to improve top-N recommendation performance.

Table 2: Comparison of the individual components of the proposed approach with the integrated model.

Method	**Beauty**	
	HR(%)	ARHR(%)
Linear Prediction	42.3 ± 0.3	37.8 ± 0.3
Self-Recovery	37.6 ± 0.3	34.4 ± 0.2
Joint Model	$\mathbf{44.8 \pm 0.4}$	$\mathbf{39.7 \pm 0.3}$
Method	**Office**	
	HR(%)	ARHR(%)
Linear Prediction	23.0 ± 0.4	15.9 ± 0.3
Self-Recovery	18.5 ± 0.4	12.6 ± 0.3
Joint Model	$\mathbf{27.6 \pm 0.5}$	$\mathbf{18.2 \pm 0.4}$
Method	**Sports&Outdoors**	
	HR(%)	ARHR(%)
Linear Prediction	46.0 ± 0.4	41.6 ± 0.3
Self-Recovery	42.6 ± 0.3	39.0 ± 0.2
Joint Model	$\mathbf{46.9 \pm 0.4}$	$\mathbf{42.3 \pm 0.4}$
Method	**Gourmet Foods**	
	HR(%)	ARHR(%)
Linear Prediction	20.6 ± 0.2	11.7 ± 0.2
Self-Recovery	13.3 ± 0.2	8.3 ± 0.2
Joint Model	$\mathbf{23.9 \pm 0.1}$	$\mathbf{13.7 \pm 0.1}$
Method	**Health**	
	HR(%)	ARHR(%)
Linear Prediction	30.4 ± 0.4	26.4 ± 0.3
Self-Recovery	27.5 ± 0.3	24.2 ± 0.2
Joint Model	$\mathbf{32.2 \pm 0.4}$	$\mathbf{27.5 \pm 0.3}$

5 Conclusion

In this paper, we proposed a novel joint discriminative prediction model for personalized top-N recommendations. The proposed model integrates information from the standard item-user purchase matrix with a linear aggregation matrix and from the auxiliary item-feature matrix with linear regression models to predict the unobserved recommendation entries. We formulated this method as a joint convex optimization problem and solved it using a projected gradient descent algorithm. We conducted experiments on five real world Amazon datasets, and the proposed approach outperforms a number of top-N recommendation methods developed in the literature. Moreover, the experimental results also demonstrated the efficacy of the novel linear regression prediction component of the proposed model, which suggests it is effective to exploit the item-based side information in a discriminative way and encourages researchers to explore prediction models for building top-N recommendation systems.

Acknowledgments

This research was supported in part by the Canada Research Chairs program.

References

[Aiolli, 2014] Fabio Aiolli. Convex AUC optimization for top-N recommendation with implicit feedback. In *RecSys*, 2014.

[Bao *et al.*, 2014] Yang Bao, Hui Fang, and Jie Zhang. TopicMF: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, 2014.

[Chen and Pan, 2013] Li Chen and Weike Pan. CoFiSet: Collaborative filtering via learning pairwise preferences over item-sets. In *SDM*, 2013.

[Cheng *et al.*, 2014] Yao Cheng, Li'ang Yin, and Yong Yu. LorSLIM: Low rank sparse linear methods for top-N recommendations. In *ICDM*, 2014.

[Christakopoulou and Karypis, 2014] Evangelia Christakopoulou and George Karypis. HOSLIM: higher-order sparse linear method for top-N recommender systems. In *PAKDD*, 2014.

[Cremonesi *et al.*, 2010] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-N recommendation tasks. In *RecSys*, 2010.

[Deshpande and Karypis, 2004] Mukund Deshpande and George Karypis. Item-based top-N recommendation algorithms. *ACM TOIS*, 22(1):143–177, 2004.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[Jain and Dhillon, 2013] Prateek Jain and Inderjit S. Dhillon. Provable inductive matrix completion. *CoRR, abs/1306.0626*, 2013.

[Kabbur *et al.*, 2013] Santosh Kabbur, Xia Ning, and George Karypis. FISM: Factored item similarity models for top-N recommender systems. In *KDD*, 2013.

[Liu and Aberer, 2014] Xin Liu and Karl Aberer. Towards a dynamic top-N recommendation framework. In *RecSys*, 2014.

[McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *RecSys*, 2013.

[Melville *et al.*, 2002] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI*, 2002.

[Mooney and Roy, 2000] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *ACM DL*, 2000.

[Ning and Karypis, 2011] Xia Ning and George Karypis. SLIM: sparse linear methods for top-N recommender systems. In *ICDM*, 2011.

[Ning and Karypis, 2012] Xia Ning and George Karypis. Sparse linear methods with side information for top-N recommendations. In *RecSys*, 2012.

[Ostuni *et al.*, 2013] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-N recommendations from implicit feedback leveraging linked open data. In *RecSys*, 2013.

[Park *et al.*, 2015] Dohyung Park, Joe Neeman, Jin Zhang, Sujay Sanghavi, and Inderjit Dhillon. Preference completion: Large-scale collaborative ranking from pairwise comparisons. In *ICML*, 2015.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[Ricci *et al.*, 2011] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[Schafer *et al.*, 2007] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, pages 291–324. Springer-Verlag, 2007.

[Sedhain *et al.*, 2016] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Darius Braziunas. On the effectiveness of linear models for one-class collaborative filtering. In *AAAI*, 2016.

[Shi *et al.*, 2012a] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. TFMAP: Optimizing map for top-N context-aware recommendation. In *SIGIR*, 2012.

[Shi *et al.*, 2012b] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys*, 2012.

[Sindhwani *et al.*, 2010] Vikas Sindhwani, Serhat Selcuk Bucak, Jianying Hu, and Aleksandra Mojsilovic. One-class matrix completion with low-density factorizations. In *ICDM*, 2010.

[Steck, 2010] Harald Steck. Training and testing of recommender systems on data missing not at random. In *KDD*, 2010.

[Verstrepen and Goethals, 2014] Koen Verstrepen and Bart Goethals. Unifying nearest neighbors collaborative filtering. In *RecSys*, 2014.

[Weimer *et al.*, 2008] Markus Weimer, Alexandros Karatzoglou, Quoc V. Le, and Alexander J. Smola. COFI RANK - maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2008.

[Xu *et al.*, 2014] Yinqing Xu, Wai Lam, and Tianyi Lin. Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups. In *CIKM*, 2014.

[Yang *et al.*, 2012] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. On top-k recommendation using social networks. In *RecSys*, 2012.

[Zhao *et al.*, 2016] Feipeng Zhao, Min Xiao, and Yuhong Guo. Predictive collaborative filtering with side information. In *IJCAI*, 2016.

[Zheng *et al.*, 2014] Yong Zheng, Bamshad Mobasher, and Robin Burke. Deviation-based contextual SLIM recommenders. In *CIKM*, 2014.