Predictive Collaborative Filtering with Side Information

Feipeng Zhao and **Min Xiao** and **Yuhong Guo**
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122, USA
{feipeng.zhao, minxiao, yuhong}@temple.edu

Abstract

Recommender systems have been widely studied in the literature as they have real world impacts in many E-commerce platforms and social networks. Most previous systems are based on the user-item recommendation matrix, which contains users' history recommendation activities on items. In this paper, we propose a novel predictive collaborative filtering approach that exploits both the partially observed user-item recommendation matrix and the item-based side information to produce top-N recommender systems. The proposed approach automatically identifies the most interesting items for each user from his or her non-recommended item pool by aggregating over his or her recommended items via a low-rank coefficient matrix. Moreover, it also simultaneously builds linear regression models from the item-based side information such as item reviews to predict the item recommendation scores for the users. The proposed approach is formulated as a rank constrained joint minimization problem with integrated least squares losses, for which an efficient analytical solution can be derived. To evaluate the proposed learning technique, empirical evaluations on five recommendation tasks are conducted. The experimental results demonstrate the efficacy of the proposed approach comparing to the competing methods.

1 Introduction

Item-based recommender systems aim to recommend new items to a target user based on the user's previous recommendation activities (*e.g.*, previous purchases, ratings, or clicks) [Sarwar *et al.*, 2001; Blei *et al.*, 2003; Deshpande and Karypis, 2004; Ostuni *et al.*, 2013]. Recommending a ranked list of new items, which may be very attractive to the user but have not been observed in the user's previous recommendation activities, can encourage additional purchase or visits, which is very important in real application scenarios. Recommender systems have been popularly studied due to their important impacts in E-commerce platforms such as Amazon and eBay, social networks such as Facebook and LinkedIn, and search engines such as Google and Bing [Ricci *et al.*, 2011]. A variety of works have been developed in the literature to address different recommender systems such as movie recommendations [Park and Pennock, 2007; Koren, 2008; Rendle *et al.*, 2009; Kabbur *et al.*, 2013], book recommendations [Ning and Karypis, 2012], music recommendations [Ning and Karypis, 2011], advert recommendations [Stern *et al.*, 2009], news recommendations [Pan *et al.*, 2008] and channel/program recommendations [Hu *et al.*, 2008].

There are many different recommendation approaches, including content filtering-based approaches [Mooney and Roy, 2000; Pazzani and Billsus, 2007] and collaborative filtering-based approaches [Sarwar *et al.*, 2001; Deshpande and Karypis, 2004; Hofmann, 2003]. The content filtering-based approaches make recommendations to a target user by only using this particular user's profile information (*e.g.*, previous recommendation activities), while the collaborative filtering-based approaches also exploit other users' profile information. Usually, these recommender systems work on the user-item recommendation matrix and generate a list of top-N potential interesting items with the highest recommendation scores for each target user from the user's non-recommended item pool. They typically retrieve relevant interesting items for target users by only using users' history recommendation activities on items, but ignoring the existing valuable side information such as the reviews for books, or the plots for the movies, which limits their recommendation capacity and performance.

In this paper, we propose a novel predictive collaborative filtering approach to generate a list of top-N items with the highest recommendation scores for each target user. The proposed approach exploits the item-based side information in addition to the conventional user-item recommendation matrix. To be more specific, it automatically identifies the recommendation scores for non-recommended items via low-rank aggregation over the users' already-recommended items to recover the full recommendation matrix for all items. Simultaneously, it incorporates the item-based side information as an item-feature matrix and trains a regularized linear regression model to predict the recommendation scores from the item features for each user. We formulate this simultaneous user-item matrix self-recovering and predictive regression model training process as a joint low-rank constrained minimization problem, and derive an efficient analytical solution for it. To evaluate the proposed approach, we conduct a set of experiments on a number of real world Amazon datasets with recommendation items such as books, beauty products, etc. The experimental results demonstrate that the proposed approach is very effective and efficient in producing top-N recommender systems, and achieves superior recommendation performance than the comparison methods.

2 Related Work

A variety of works have been developed in the literature to address item-based recommender systems. We provide a brief review below on the most related collaborative filtering approaches and side information assisted approaches.

Collaborative filtering approaches make recommendations by using both the target user's previous recommendation activities and other users' history behaviors [Su and Khoshgoftaar, 2009], which include two groups of methods, neighborhood-based methods [Sarwar *et al.*, 2001; Deshpande and Karypis, 2004; Kabbur *et al.*, 2013] and latent factor-based methods [Blei *et al.*, 2003; Hofmann, 2003; Rennie and Srebro, 2005; Salakhutdinov *et al.*, 2007; Weimer *et al.*, 2007; Koren *et al.*, 2009; Cremonesi *et al.*, 2010; Sindhwani *et al.*, 2010].

Neighborhood-based collaborative filtering methods make recommendations by exploiting similarities among items within the user-item recommendation matrix. Once item-item similarities are computed, the prediction on a new item for a user is obtained by considering the particular user's past recommendation activities on similar items. For example, the work in [Sarwar *et al.*, 2001] proposed to use similarity-weighted sum prediction to recommend new items for a target user. In order to measure the item-item similarities, various similarity schemes have been investigated, including cosine similarity, correlation similarity, adjusted cosine similarity [Sarwar *et al.*, 2001], and conditional probability similarity [Deshpande and Karypis, 2004].

Unlike the neighborhood-based methods, which directly employ the user-item recommendation matrix for prediction, the latent factor-based collaborative filtering methods first identify a set of latent factors for users and items respectively. They then train a prediction model on the latent space to make recommendations. In order to identify the latent factors, a variety of models have been employed, including the pure singular value decomposition (PureSVD) method [Cremonesi *et al.*, 2010], the Gaussian probabilistic latent semantic analysis method [Hofmann, 2003], the latent Dirichlet allocation method [Blei *et al.*, 2003], the matrix factorization method [Koren *et al.*, 2009], the maximum margin matrix factorization methods [Rennie and Srebro, 2005; Weimer *et al.*, 2007], the pairwise comparison method [Rendle *et al.*, 2009], the restricted Boltzmann machines [Salakhutdinov *et al.*, 2007], and the weighted non-negative matrix factorization method [Sindhwani *et al.*, 2010].

In addition to these two groups, some other collaborative filtering recommender systems have also been developed in the literature [Koren, 2008; Ning and Karypis, 2011; Shi *et al.*, 2012]. Koren (2008) proposed an asymmetric-SVD based hybrid approach to address recommendation problems, which combines the similarity-based neighborhood method and the latent factor-based method together for recommendations. Ning and Karypis (2011) proposed a sparse linear method to address top-N recommendation problems, which computes the recommendation score on a non-purchased/non-recommended item for a target user using sparse aggregation of purchased/recommended items of the particular user. Shi *et al.* (2012) proposed a collaborative less-is-more filtering approach, which maximizes the mean reciprocal rank to address recommendation problems in social networks.

Different from collaborative filtering approaches, which only exploit previous recommendation activities in user-item matrix, *side information assisted approaches* also exploit additional meta-data information (*e.g.*, product reviews, film plots, etc) to recommend new items for a target user [Stern *et al.*, 2009; Gunawardana and Meek, 2009; de Campos *et al.*, 2010; Ning and Karypis, 2012]. Stern *et al.* (2009) proposed to use both user meta-data such as a user's job, age, gender and id, and item meta-data such as a movie's genre and id. They represented these meta-data as feature vectors for the users and items respectively and incorporated the meta-data information into the collaborative filtering recommender system. The work in [Gunawardana and Meek, 2009] incorporated both the user-item recommendation information and the item meta-data into a unified Boltzmann machine to form a hybrid recommender system. Similarly, the work in [de Campos *et al.*, 2010] developed a hybrid recommender system based on Bayesian networks over user nodes, item nodes, and item-feature nodes. The approach propagates content-based information from recommended items to non-recommended items via the item-feature nodes, and propagates collaborative-based information via item nodes. Ning and Karypis (2012) proposed to incorporate item-based side information into the sparse linear method [Ning and Karypis, 2011] to learn the aggregation coefficient matrix and address top-N recommendation problems, which has demonstrated superior performance over conventional recommender systems without side information. Different from these previous methods, our proposed approach performs a joint predictive recovery of the user-item matrix for top-N recommendations by integrating the partially observed user-item matrix and the item-based side information such as reviews in a discriminative and mutually complementary manner.

A few recent works [Bao *et al.*, 2014; McAuley and Leskovec, 2013; Xu *et al.*, 2014] have exploited user reviews for building recommender systems, while [Chen *et al.*, 2015] provides a survey of such recent attempts. However most of these methods require the reviews to be directly associated with the user ratings, while our approach treats reviews as side information and can exploit reviews from auxiliary resources that have no direct relationships with the target users.

3 Proposed Approach

In this section, we present the proposed predictive collaborative filtering approach for item-based top-N recommender systems. We assume an observed item-user recommendation matrix $Y \in \mathbb{R}^{n\times t}$, which contains the observed recommendation (or purchase) activities of t users over n product items. In this matrix, an observed entry Y_{ij} (with an indicator value 1 or a positive recommendation score) indicates the i-th product item has been recommended (or purchased) by the j-th user, while an unobserved entry $Y_{ij} = 0$ only indicates an unknown relationship between the j-th user and the i-th product item, i.e., the recommendation (or purchase) action of the j-th user over the i-th item has not been observed in the user's past recommendation activities, but it is possible that the j-th user can be interested in the i-th item. As item-based side information in the form of product reviews for E-commerce applications or film plots for movie recommendations exists in many application domains, we take the item-based side information into account as well. In particular, we assume there is an item-feature matrix $X \in \mathbb{R}^{n\times d}$, whose each row contains a feature vector for the corresponding product item and presents the item-based side information. We aim to automatically identify the most promising interesting product items, in particular the top-N items, for each target user from his non-recommended items by using both the observed item-user recommendation matrix Y and the item-feature matrix X. Below we first present a matrix self-completion component based on the item-user recommendation matrix, and then incorporate the predictions from the item-feature matrix to form a joint prediction model. We finally derive a convenient analytical solution for the proposed joint learning method.

In the rest of the paper, we use $\mathbf{1}$ to denote a column vector of all 1s, while assuming the size of the column vector can be inferred from the context. We use I_n to denote an identity matrix with size n. For any matrix X, $\|X\|_F$ denotes its Frobenius norm, and $X^\dagger$ denotes its pseudo-inverse.

3.1 A Joint Prediction Model

Given the past recommendation/purchase activities in the observed item-user recommendation matrix Y, the recommendation score on a non-recommended/non-purchased item v_i for a user u_j can be calculated as a linear aggregation of the items that have already been recommended/purchased by u_j, that is, $\widetilde{Y}_{ij} = \mathbf{w}^\top Y_{:j}$, where $Y_{:j}$ denotes the j-th column of Y. Such a linear recovery model can be built for all items with $\widetilde{Y}_{:j} = WY_{:j}$, while the linear functions have to be learned from the statistical item aggregation relationships existed in the observed data for all the users. This leads to a matrix self-recovery (self-completion) formulation $\widetilde{Y} = WY$, where W is an $n \times n$ linear aggregation coefficient matrix that is shared across all the t users. A meaningful matrix self-recovery should automatically complete the unobserved recommendation scores in the matrix without affecting much the observed recommendation scores. We hence enforce the self-recovered item-user matrix to maximally preserve the original matrix values and formulate the self-recovery model as the following learning problem

$$\min_{W} \quad \|Y - WY\|_F^2 \qquad \text{s.t.} \quad \text{rank}(W) \leq \lambda \qquad (1)$$

To avoid the trivial solution of setting W as an identity matrix, we assume the coefficient matrix W should be low rank and encode this low rank property using a rank constraint with a hyperparameter λ. This low rank property can also allow the model to implicitly exploit the latent low-dimensional representations of the items.

In addition to the item-user recommendation matrix Y, we also have explicit item-based side information available as an item-feature matrix X, where each item feature vector describes the properties of a given product item. Note that this product item side information is common to all users. But different users may have different tastes over the same product item. The recommendation scores of each user over the set of product items depend on the properties of the product items. Hence we propose to predict the recommendation score of each user over a product item from the feature vector of this product item. In particular, we can use a linear regression model, $f_j(\mathbf{x}) = \mathbf{x}^\top \mathbf{q} + b$, for the j-th user to predict his or her recommendation score y from the item feature vector $\mathbf{x}$. Given the observed item-feature matrix X and the recovered item-user recommendation matrix $\widetilde{Y}$, the linear regression model f_j for the j-th user can be trained on the data by minimizing a least squares regression loss $\|X\mathbf{q} + b\mathbf{1} - \widetilde{Y}_{:j}\|^2$ over the model parameters $\mathbf{q}$ and b. Thus the linear regression model is specifically tailored to the j-th user and can help to capture user-specific preferences of the product items when making recommendations. For all the t users, we will train t linear regression models from the data by minimizing the following regularized least squares loss function

$$\min_{Q,b} \quad \|XQ + \mathbf{1}\mathbf{b}^\top - \widetilde{Y}\|_F^2 + \beta\|Q\|_F^2 \qquad (2)$$

where $Q \in \mathbb{R}^{d\times t}$ and $\mathbf{b} \in \mathbb{R}^t$ are the model parameters for the t linear regression models, such that $f_j(\mathbf{x}) = \mathbf{x}^\top Q_{:j} + \mathbf{b}_j$ for the j-th user.

Since $\widetilde{Y} = WY$ is a self-recovered matrix via Eq. (1), by integrating the recommendation matrix recovery in Eq. (1) and the linear regression model training in Eq. (2) together, we formulate a joint recommendation prediction model as follows to perform joint recommendation score prediction in a mutually complementary manner

$$\min_{W,Q,\mathbf{b}} \quad \|XQ + \mathbf{1}\mathbf{b}^\top - WY\|_F^2 + \beta\|Q\|_F^2 + \gamma\|Y - WY\|_F^2$$
$$s.t. \quad \text{rank}(W) \leq \lambda \qquad (3)$$

where β and γ are the trade-off parameters. By solving this joint minimization problem, we expect to integrate both the existing recommendation activity information and the item-based side information to predict the recommendation scores of each user over his or her non-recommended/non-purchased items. The top-N recommendations for a user u_j can be produced by sorting u_j's non-recommended/non-purchased items in a decreasing order based on their predicted recommendation scores.

3.2 Learning Algorithm

The learning problem in Eq. (3) is a joint minimization problem over both the linear regression model parameters $\{Q, \mathbf{b}\}$ and the linear aggregation coefficient matrix W. We first perform minimization over the linear regression model parameters assuming fixed W, which produces the following closed-form solutions for Q and $\mathbf{b}$,

$$Q = \left(X^\top HX + \beta I_d\right)^{-1} X^\top HWY \qquad (4)$$

$$\mathbf{b} = \frac{1}{n}\left(WY - XQ\right)^\top \mathbf{1} \qquad (5)$$

Algorithm 1 Learning Algorithm

Input: X, Y, $\beta > 0$, $\gamma > 0$, $\lambda > 0$.
Procedure:

1. Compute auxiliary matrices F and S with Eq. (7)
2. Perform singular value decomposition over F and Y, compute $\mathcal{P}_{F,\mathcal{L}}$ and $\mathcal{P}_{Y,\mathcal{R}}$, and let $M = \mathcal{P}_{F,\mathcal{L}} S \mathcal{P}_{Y,\mathcal{R}}$.
3. Compute a solution W^* using Eq. (9), and set $\widetilde{Y}^* = W^* Y$.

where $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is a centering matrix.

By plugging the solutions for the regression model parameters Q in Eq. (4) and $\mathbf{b}$ in Eq. (5) back into the objective function in Eq. (3), the joint learning problem in Eq. (3) can be equivalently reformulated into the following minimization problem over W,

$$\min_{W:\,\text{rank}(W)\leq\lambda} \quad \|BWY\|_F^2 + \beta\|AWY\|_F^2 + \gamma\|Y - WY\|_F^2 \tag{6}$$

where $A = (X^\top HX + \beta I_d)^{-1}X^\top H$ and $B = H\,(XA - I_n)$. By further introducing the following replacement matrices,

$$F = \begin{bmatrix} B \\ \sqrt{\beta}A \\ \sqrt{\gamma}I_n \end{bmatrix}, \quad S = \begin{bmatrix} O_{(n+d)\times t} \\ \sqrt{\gamma}Y \end{bmatrix} \tag{7}$$

where $O_{(n+d)\times t}$ denotes a $(n+d)\times t$ matrix with all 0s, the problem (6) can be rewritten as

$$\min_{W:\,\text{rank}(W)\leq\lambda} \quad \|FWY - S\|_F^2 \tag{8}$$

Though the optimization problem in (8) is still a non-convex optimization problem due to the rank constraint, below we will derive a convenient analytical solution for it.

For matrix F, we denote its thin SVD as $F = U_F\Sigma_F V_F^\top$. Two projections can be defined consequently as $\mathcal{P}_{F,\mathcal{L}} := U_F U_F^\top$ and $\mathcal{P}_{F,\mathcal{R}} := V_F V_F^\top$. Similarly, we can define projections $\mathcal{P}_{Y,\mathcal{L}}$ and $\mathcal{P}_{Y,\mathcal{R}}$ for matrix Y. We then have the following theorem.

Theorem 1 *For $\mathbb{N} \ni \lambda \leq$ rank$(\mathcal{P}_{F,\mathcal{L}} S \mathcal{P}_{Y,\mathcal{R}})$, let $M = \mathcal{P}_{F,\mathcal{L}} S \mathcal{P}_{Y,\mathcal{R}}$, then*

$$W^* = F^\dagger M_{(\lambda)} Y^\dagger \tag{9}$$

is a solution to the minimization problem in (8), where the operation $M_{(\lambda)}$ extracts the truncation of M by setting all of its singular values as zeros except the λ largest ones, i.e.,

$$M = U_M\Sigma_M V_M^\top, \quad M_{(\lambda)} = U_M\widehat{\Sigma}_M(\lambda)V_M^\top, \tag{10}$$

and $\widehat{\Sigma}_M(\lambda)$ is the truncation of Σ by keeping only the largest λ diagonal values of Σ and setting all others as zeros. The solution is unique iff the λ-th and $(\lambda+1)$-th largest singular values of M are different.

Theorem 1 was proved by Sondermann [Sondermann, 1986] and Friedland&Torokhti [Friedland and Torokhti, 2007]. It provides an analytical closed-form solution for our rank constrained minimization problem (8). The overall learning algorithm is summarized in Algorithm 1.

Table 1: Statistic summary for the experimental datasets.

Dataset	#items	#users	#trns	rsize	csize	density
Beauty	7034	5294	23576	4.45	3.35	0.063%
Office	6167	2389	10988	4.60	1.78	0.075%
Sports&Outdoors	7085	7971	32742	4.10	4.62	0.058%
Gourmet Foods	8322	4780	23553	4.93	2.83	0.059%
Health	7877	9121	38215	4.19	4.85	0.053%

The complexity of Algorithm 1 is dominated by the three singular value decomposition operations on F, Y and M respectively. This leads to a computational complexity of $O\left((2n+d)n^2 + \min(n^2t, nt^2) + \min((2n+d)^2t, (2n+d)t^2)\right)$ for the proposed algorithm when a standard full SVD algorithm is used [Brand, 2006]. Moreover, based on the definitions of F and S in Eq.(7) and the definition of M in Theorem 1, it is easy to conclude that rank$(F) = n$ and rank$(M) \leq$ rank(S)=rank(Y). The item-user recommendation matrix Y is typically low rank due to the existence of dependencies and correlations among the users' activities. Hence if r=rank(Y), one could have $r \ll \min(n,t)$. In this case, the computational complexity of Algorithm 1 can be further reduced to $O\left((2n+d)n^2 + r(3n+d)t\right)$ by using a fast SVD algorithm in [Brand, 2006].

After obtaining the solution W^* in Eq. (9) for the coefficient matrix, we can get the recovered item-user recommendation matrix $\widetilde{Y}^*$ by setting $\widetilde{Y}^* = W^*Y$, which contains the predicted recommendation scores for the originally non-recommended/non-purchased items of each user. We then perform top-N recommendation in the following way. For the j-th user, $\widetilde{Y}^*_{:j}$ denotes his predicted recommendation scores over all the product items. We rank the scores for non-recommended items in a non-increasing order and consider the first N items as the predicted recommendation items.

4 Experiments

In this section, we present our empirical evaluations and discussions of recommender systems on real-world datasets.

Datasets

We conducted experiments on five real-world Amazon datasets [McAuley and Leskovec, 2013], *Beauty*, *Office*, *Sports&Outdoors*, *Gourmet Foods* and *Health*, each of which corresponds to one category of Amazon products. For each category, the original dataset contains transactions between different product items and users, indicated with non-zero multivariate rating values. We converted the multivariate rating values to 1s and filtered those less popular product items and users that appeared less than three transactions to construct the item-user recommendation matrix. The statistic summary of the produced datasets is presented in Table 1. For each dataset, we present the number of items (#items), the number of users (#users), the number of transactions (#trns), the average number of transactions for each item (#rsize), the average number of transactions for each user (#csize) and the density value (#trns/(#users×#items)).

Item-based Side Information

For each dataset, we used the product reviews as the item-based side information for the product items. We extracted

Table 2: Average performance comparison in terms of HR, ARHR, model learning time (mt) and test time (tt) in seconds. The *params* columns present the parameter setting for each approach. *PureSVD* has one parameter f, indicating the number of latent factors; *BPRMF* has two parameters, latent factor dimension f and regularization parameter λ; *SLIM* has two parameters, the ℓ_2-norm and ℓ_1-norm regularization parameters β and λ; *cSLIM* has three parameters, the side-information weighting parameter α, the ℓ_2-norm and ℓ_1-norm regularization parameters β and λ; the proposed approach has three parameters, β, γ, λ.

feature	method		**Beauty** params		HR	ARHR	mt	tt
F_{no}	PureSVD	100	-	-	0.214 ± 0.005	0.178 ± 0.005	112s	0.5s
	BPRMF	500	0.01	-	0.264 ± 0.001	0.234 ± 0.002	1900s	0.6s
	SLIM	10	0.01	-	0.289 ± 0.001	0.258 ± 0.001	3500s	5s
F_{tf}	cSLIM	1e-6	10	0.01	0.299 ± 0.001	0.263 ± 0.001	38000s	5s
	Proposed	200	0.01	2000	0.346 ± 0.001	0.284 ± 0.002	1800s	10s
feature	method		**Office** params		HR	ARHR	mt	tt
F_{no}	PureSVD	100	-	-	0.095 ± 0.005	0.075 ± 0.005	21s	0.3s
	BPRMF	200	0.01	-	0.132 ± 0.003	0.095 ± 0.001	800s	0.1s
	SLIM	100	0.001	-	0.166 ± 0.003	0.110 ± 0.002	1200s	2s
F_{tf}	cSLIM	1e-6	100	0.001	0.177 ± 0.002	0.111 ± 0.002	2800s	2s
	Proposed	100	0.1	2000	0.219 ± 0.001	0.151 ± 0.001	600s	6s
feature	method		**Sports&Outdoors** params		HR	ARHR	mt	tt
F_{no}	PureSVD	500	-	-	0.154 ± 0.005	0.126 ± 0.005	200s	2s
	BPRMF	500	0.01	-	0.186 ± 0.002	0.158 ± 0.001	3500s	0.3s
	SLIM	10	0.01	-	0.221 ± 0.001	0.185 ± 0.001	4300s	7s
F_{tf}	cSLIM	1e-6	10	0.01	0.224 ± 0.001	0.187 ± 0.001	46000s	5s
	Proposed	200	0.05	2000	0.238 ± 0.001	0.187 ± 0.001	1000s	10s
feature	method		**Gourmet Foods** params		HR	ARHR	mt	tt
F_{no}	PureSVD	500	-	-	0.069 ± 0.005	0.045 ± 0.005	150s	0.5s
	BPRMF	200	0.01	-	0.076 ± 0.002	0.053 ± 0.001	2100s	0.4s
	SLIM	10	0.01	-	0.122 ± 0.002	0.074 ± 0.001	4800s	6s
F_{tf}	cSLIM	1e-6	10	0.01	0.129 ± 0.002	0.081 ± 0.001	52000s	5s
	Proposed	200	0.05	2000	0.173 ± 0.001	0.089 ± 0.001	1400s	9s
feature	method		**Health** params		HR	ARHR	mt	tt
F_{no}	PureSVD	100	-	-	0.122 ± 0.005	0.095 ± 0.003	250s	2s
	BPRMF	500	0.01	-	0.162 ± 0.001	0.121 ± 0.001	4500s	1s
	SLIM	1	0.01	-	0.191 ± 0.001	0.159 ± 0.001	8000s	12s
F_{tf}	cSLIM	1e-7	1	0.01	0.209 ± 0.001	0.155 ± 0.001	70000s	10s
	Proposed	200	0.05	2000	0.211 ± 0.001	0.161 ± 0.001	1400s	14s

the unigram features from the review articles with stopwords being removed. We further selected the most frequent 5000 unigram features as the item features and represented each product item as a bag-of-word feature vector with term-frequency feature values (F_{tf}). We denote the recommender systems without item-based side information as F_{no}.

Comparison Approaches

We compared our proposed approach with the following approaches for recommender systems: (1) Pure Singular Value Decomposition (*PureSVD*), which is a baseline collaborative filtering method based on singular value decomposition [Cremonesi *et al.*, 2010]; (2) Bayesian Personalized Ranking for matrix factorization (BPRMF), which exploits pairwise comparison [Rendle *et al.*, 2009]; (3) Sparse Linear Model (*SLIM*), which learns a sparse coefficient matrix to induce recommendation scores [Ning and Karypis, 2011]; and (4) Collective Sparse Linear Model (*cSLIM*), which incorporates item-based side information into the SLIM model to generate the top-N recommendations [Ning and Karypis, 2012]. These methods address collaborative filtering from different aspects and have demonstrated effective performance in the literature.

Evaluation Criteria

We used the commonly used leave-one-out cross validation to evaluate the performance of each recommendation algorithm over five runs. For each run, we randomly chose one transaction for each user and put it into the test set. The training set is the masked item-user transaction matrix with the transactions of the test set removed. We ran each recommender system five times based on random partitions of the training and test sets. The evaluation is conducted on the top-N items recommended for each user by the recommender system. We set N=10 and used the following two criteria to evaluate the top-N recommendation performance: the *Hit Rate (HR)* and the *Average Reciprocal Hit-Rank (ARHR)* [Deshpande and

Karypis, 2004],

$$\text{HR} = \frac{\#\text{hits}}{\#\text{users}}, \qquad \text{ARHR} = \frac{1}{\#\text{users}} \sum_{i=1}^{\#\text{hits}} \frac{1}{p_i} \tag{11}$$

where #users is the number of users, #hits is the number of users with the test item recommended in the top-N list, and p_i is the position of the item in the top-N ranked list. Larger HR and ARHR values indicate better performance.

4.1 Experimental Results

We compared the five comparison methods (*PureSVD, BPRMF, SLIM, cSLIM, Proposed*) on the five datasets using the evaluation criteria presented above. The average comparison results for all these methods in terms of HR and ARHR are presented in Table 2. We can see that the baseline *PureSVD* method does not work well on all the datasets comparing to the other systems, which shows that the pure singular value decomposition-based collaborative filtering method is far from enough to develop a good recommender system without using other information. The *BPRMF* and *SLIM* methods on the other hand work much better than the *PureSVD* method on all the datasets in both measurements. The *cSLIM* method, which utilizes item-based side information, outperforms *PureSVD, BPRMF* and *SLIM* across all cases except on *Health* in terms of *ARHR*. This shows that the item-based side information is useful for improving the recommendation performance. Our proposed approach on the other hand consistently produced the best results among all the comparison methods on all the five datasets in terms of both measurements. This demonstrates that our proposed approach provides an effective mechanism to integrate information from the item-user matrix and the item-based side information for producing top-N recommender systems.

Moreover, we also recorded the model learning time and the recommendation test time for all the comparison approaches in all these experiments. These running times are reported in Table 2 as well. We can see that with an analytical solution our proposed method is very efficient in learning. It significantly reduced the training time by comparing not only to the *cSLIM* method which also utilizes item-based side information, but also to *SLIM* and *BPRMF* that are not able to incorporate side information. On the datasets with larger recommendation matrices such as *Gourmet Foods* and *Health*, the differences between the model learning times of the proposed method and the *cSLIM* method are quite dramatic. Though the *PureSVD* method runs faster than our proposed approach, this baseline cannot exploit side information and produced very poor performance. Our proposed approach on the other hand maintains both efficiency and effectiveness. All these results demonstrate that the proposed approach is an effective and efficient state-of-the-art solution for top-N recommender systems.

4.2 Integration Model Study

To verify the effectiveness of the integration mechanism of the proposed joint model, we also compared the two individual components, the self-completion component and the side-information based prediction component, to the proposed joint model in terms of top-N recommendation performance. The comparison results in terms of *HR* and *ARHR* are reported in Table 3. We can see that the proposed joint model consistently outperforms both of the individual components. With each component alone, the model cannot achieve a state-of-the-art performance. This suggests that our proposed model provides an effective integration over the individual information sources.

Table 3: Comparisons of the side-information based prediction component, self-completion component, and the proposed joint model.

Method	**Beauty** HR	**Beauty** ARHR
Side-Prediction	0.048 ± 0.000	0.022 ± 0.000
Self-Completion	0.286 ± 0.002	0.256 ± 0.002
Proposed	0.346 ± 0.002	0.284 ± 0.001
Method	**Office** HR	**Office** ARHR
Side-Prediction	0.057 ± 0.000	0.019 ± 0.000
Self-Completion	0.152 ± 0.001	0.102 ± 0.001
Proposed	0.219 ± 0.001	0.151 ± 0.001
Method	**Sports&Outdoors** HR	**Sports&Outdoors** ARHR
Side-Prediction	0.034 ± 0.000	0.011 ± 0.000
Self-Completion	0.216 ± 0.001	0.183 ± 0.001
Proposed	0.238 ± 0.001	0.187 ± 0.001
Method	**Gourmet Foods** HR	**Gourmet Foods** ARHR
Side-Prediction	0.048 ± 0.000	0.022 ± 0.000
Self-Completion	0.113 ± 0.001	0.070 ± 0.001
Proposed	0.173 ± 0.001	0.089 ± 0.001
Method	**Health** HR	**Health** ARHR
Side-Prediction	0.036 ± 0.000	0.0138 ± 0.000
Self-Completion	0.185 ± 0.003	0.154 ± 0.002
Proposed	0.211 ± 0.001	0.161 ± 0.001

5 Conclusion

In this paper, we presented a novel joint prediction approach that exploits both the conventional user-item recommendation matrix and the item-based side information in a complementary manner to generate top-N recommendations for target users. We formulated the simultaneous user-item matrix recovering and item-based linear regression model training process as a joint low-rank constrained minimization problem, and derived an analytical closed-form solution for it. We conducted a set of experiments on five real world Amazon datasets with different recommendation items. The experimental results demonstrated that the proposed approach is effective and efficient in producing top-N recommender systems, and outperforms the comparison methods.

Acknowledgments

This research was supported in part by NSF IIS-1422127.

References

[Bao *et al.*, 2014] Y. Bao, H. Fang, and J. Zhang. TopicMF: Simultaneously exploiting ratings and reviews for recommendation. In *Proc. of AAAI*, 2014.

[Blei *et al.*, 2003] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[Brand, 2006] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415:20–30, 2006.

[Chen *et al.*, 2015] L. Chen, G. Chen, and F. Wang. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Inter*, 25:99–154, 2015.

[Cremonesi *et al.*, 2010] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. of RecSys*, 2010.

[de Campos *et al.*, 2010] L. de Campos, J. Fernández-Luna, J. Huete, and M. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *Intern. J. of Approximate Reason.*, 51(7):785–799, 2010.

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

[Friedland and Torokhti, 2007] S. Friedland and A. Torokhti. Generalized rank- constrained matrix approximations. *SIAM J. on Mat. Anal. and App.*, 29(2):656–659, 2007.

[Gunawardana and Meek, 2009] A. Gunawardana and C. Meek. A unified approach to building hybrid recommender systems. In *Proc. of RecSys*, 2009.

[Hofmann, 2003] T. Hofmann. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *ACM SIGIR Conf. on Res. and Develop. in Info. Retrieval*, 2003.

[Hu *et al.*, 2008] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM*, 2008.

[Kabbur *et al.*, 2013] S. Kabbur, X. Ning, and G. Karypis. FISM: factored item similarity models for top-N recommender systems. In *Proc. of KDD*, 2013.

[Koren *et al.*, 2009] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Koren, 2008] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. of KDD*, 2008.

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

[Mooney and Roy, 2000] R. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proc. of Conf. on Digit. Libraries*, 2000.

[Ning and Karypis, 2011] X. Ning and G. Karypis. SLIM: sparse linear methods for top-N recommender systems. In *Proc. of ICDM*, 2011.

[Ning and Karypis, 2012] X. Ning and G. Karypis. Sparse linear methods with side information for Top-N recommendations. In *Proc. of RecSys*, 2012.

[Ostuni *et al.*, 2013] V. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. Top-N recommendations from implicit feedback leveraging linked open data. In *RecSys*, 2013.

[Pan *et al.*, 2008] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proc. of ICDM*, 2008.

[Park and Pennock, 2007] S. Park and D. Pennock. Applying collaborative filtering techniques to movie search for better ranking and browsing. In *Proc. of ICDM*, 2007.

[Pazzani and Billsus, 2007] M. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer-Verlag, 2007.

[Rendle *et al.*, 2009] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI*, 2009.

[Rennie and Srebro, 2005] J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML*, 2005.

[Ricci *et al.*, 2011] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. Springer, 2011.

[Salakhutdinov *et al.*, 2007] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proc. of ICML*, 2007.

[Sarwar *et al.*, 2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*, 2001.

[Shi *et al.*, 2012] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proc. of RecSys*, 2012.

[Sindhwani *et al.*, 2010] V. Sindhwani, S. Bucak, J. Hu, and A. Mojsilovic. One-class matrix completion with low-density factorizations. In *Proc. of ICDM*, 2010.

[Sondermann, 1986] D. Sondermann. Best approximate solutions to matrix equations under rank restrictions. *Statist. Papers*, 27:57–66, 1986.

[Stern *et al.*, 2009] D. Stern, R. Herbrich, and T. Graepel. Matchbox: Large scale online Bayesian recommendations. In *Proc. of WWW*, 2009.

[Su and Khoshgoftaar, 2009] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

[Weimer *et al.*, 2007] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. $\text{COFI}^{\text{RANK}}$ - maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

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