Top-N Recommendation with High-Dimensional Side Information via Locality Preserving Projection

Yifan Chen
National University of Defense Technology
Changsha, China
yfchen@nudt.edu.cn

Xiang Zhao
National Univ. of Defense Tech.
Collaborative Innovation Center of Geospatial Technology
Changsha, China
xiangzhao@nudt.edu.cn

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

In this paper, we leverage high-dimensional side information to enhance top-N recommendations. To reduce the impact of the curse of high dimensionality, we incorporate a dimensionality reduction method, Locality Preserving Projection (LPP), into the recommendation model. A joint learning model is proposed to achieve the task of dimensionality reduction and recommendation simultaneously and iteratively. Specifically, item similarities generated by the recommendation model are used as the weights of the adjacency graph for LPP while the projections are used to bias the learning of item similarity. Employing LPP for recommendation not only preserves locality but also improves item similarity. Our experimental results illustrate that the proposed method is superior over state-of-the-art methods.

1 INTRODUCTION

Top-N recommendation has been widely adopted to recommend *ranked lists of items* so as to help users identify the items that best fit their personal tastes. Over the last decades, various efforts have been dedicated to provide top-N recommendations. Among them, the *item*-based scheme stands out for its solid performance. Representative methods include item-based k-nearest-neighbor, sparse linear methods (SLIM) [5], and so forth, which have been shown to outperform *user*-based scheme.

The recommendation accuracy of such item-based neighborhood methods relies largely on the item similarities computed or learned. Specifically, item similarities are usually made available based on user feedback (both explicit and implicit), e.g., purchases, ratings, reviews, clicks, and check-ins. Lately, there has been an increase in the amount of additional information associated with items, referred to as *side information* [6]. Typical examples include descriptions of movies in movie recommendation, resumes of applicants in job matching, content of emails in spam detection, reviews of items in online shopping, and so forth. Side information has generated the interest of many researchers and has led to the development of *hybrid* algorithms to enhance the performance of recommendations by taking advantage of such information.

Side information comes with a *high dimensionality*. For example, side information can be the text descriptions of items; when regarding each unique term in the corpus as one dimension, it is indisputably high-dimensional. Moreover, side information can also be in the form of images or videos where the dimensionality is evidently much higher. Nonetheless, existing methods overlook this fact when utilizing side information, and hence, they are facing problems of efficiency and accuracy due to the curse of high dimensionality. We address the issue in this paper, and investigate how to leverage side information to boost the recommendation performance while limiting the impact from high dimensionality.

While side information is high-dimensional and sparse, it is reasonable to expect a low dimensionality of intrinsic features, and this suggests that we should incorporate dimensionality reduction for this task. Among the many available dimensionality reduction methods, Locality Preserving Projection (LPP) [3] has been shown to produce a low-dimensional space that well preserves locality. As recommendation quality largely depends on item similarity, LPP is a natural candidate in this setting.

To summarize, we propose a top-N recommendation method to harness high-dimensional side information. By introducing a projection matrix, high-dimensional side information is reduced into a low-dimensional space. We present a joint learning model to simultaneously perform LPP and learn item similarity. We then conceive an alternative iterative optimization method to solve the model. Our experimental evaluation shows that the proposed method enjoys a performance gain of up to 21.2% on Hit Rate and 36.8% on Average Reciprocal Hit Rate over state-of-the-art methods.

2 RELATED WORK

We are aware of several recent methods that leverage side information for top-N recommendation. On top of SLIM [5], SSLIM [6] utilizes a regularized optimization process to learn a sparse coefficient matrix. UFSM [1] combines item similarity model with factor models. Recently, Zhao et al. [9] have proposed a predictive collaborative filtering approach to utilize side information.

We also summarize recent methods using side information for rating prediction. AFM [2] maps side information to latent item factors by learning the map function. LCE [8] proposes a local collective factorization method. Lu et al. [4] propose an interactive

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan.

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080697

model for matrix completion. Distinct from them, we integrate dimensionality reduction into top-N recommendation.

As to dimensionality reduction, this topic has been investigated extensively, for sparse feedback via various methods [7], including principal component analysis, singular value decomposition, non-negative matrix factorization and so on. However, high-dimensional side information has rarely been addressed in the setting, and this paper tries to fill in the gap.

3 THE PROPOSED APPROACH

3.1 Notation

We first introduce the notations used throughout the paper. Let U and I be the sets of all users and all items, respectively, each of size m and n. The user feedback (both explicit and implicit) shows the items that the users have purchased, viewed or rated, which is denoted by a matrix $\boldsymbol{R}$ of size $m \times n$. We treat feedback as binary, that is, if user u provided feedback for item i, then the (u, i)-entry of $\boldsymbol{R}$ (denoted by r_{ui}) is 1, otherwise it is 0. The item similarity matrix is represented by $\boldsymbol{S} \in \mathbb{R}^{n\times n}$, where each value of entry s_{ij} is within $[0, 1]$. The feature matrix (side information associated with items) is denoted by $\boldsymbol{F} \in \mathbb{R}^{n\times d}$, where d indicates the dimensionality of side information. The projection matrix is denoted by $\boldsymbol{W} \in \mathbb{R}^{d\times k}$, which is used to map d-dimensional side information into a k-dimensional space where $k \ll d$.

3.2 Model description

This section describes the proposed model. We start with introducing the Baseline method without performing dimensionality reduction, then summarize LPP, and explain how to incorporate it in a recommender system. Finally, the proposed method is formed.

Recommendation with side information. Typically, top-N recommender systems perform matrix completion for $\boldsymbol{R}$, the core of which is to learn item similarity, which is directly relevant to recommendation. Side information is utilized to enhance the learning of item similarity. While various forms of incorporating side information exist, we incorporate a regularization term on $\boldsymbol{S}$ along with feature matrix $\boldsymbol{F}$ and form the model as the following problem:

$$\frac{1}{2}\|\boldsymbol{R} - \boldsymbol{RS}\|_F^2 + \frac{\alpha}{2}\sum_{i,j}^{n}\left(\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2 s_{ij}\right) + \frac{\lambda}{2}\|\boldsymbol{S}\|_F^2, \tag{1}$$

such that $\boldsymbol{s}_j^T\boldsymbol{1} = 1, \forall j = 1, \cdots, n$; $0 \le s_{ij} \le 1, \forall i, j = 1, \cdots, n$; and $s_{jj} = 0, \forall j = 1, \cdots, n$. $\boldsymbol{s}_j$ is the j-th column vector of $\boldsymbol{S}$, representing how similar item j is to other items. The constraint $\boldsymbol{s}_j^T\boldsymbol{1} = 1$ is incorporated to avoid the case when the learned $\boldsymbol{S}$ is close to $\boldsymbol{0}$ especially when $\boldsymbol{R}$ is very sparse. The term $\frac{1}{2}\|\boldsymbol{R}-\boldsymbol{RS}\|_F^2$ in the objective function tries to reconstruct the feedback matrix by learning the coefficient matrix $\boldsymbol{S}$, which was first introduced by SLIM [5] for top-N recommendation. As suggested there, the ℓ_2-norm is used to regularize $\boldsymbol{S}$. While ℓ_1-norm is also suggested to encourage sparsity, it is omitted as it turns out to be constant here (due to $\boldsymbol{s}_j^T\boldsymbol{1} = 1$). α is a user-specified parameter to balance the two sources of information. We further justify the regularization to $\boldsymbol{S}$ by $\boldsymbol{F}$ in detail. Given the feature matrix $\boldsymbol{F}$, $\boldsymbol{f}_i$ represents the feature vector for item i. A natural way to measure the item distance in terms of features is to compute the Euclidean distance between them, i.e., $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2$. Although the item similarity is unknown, it is reasonable to assume that closer items (in terms of feature distance) are likely to have higher similarities, and thus, item similarity between item i and j can be regularized as $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2 s_{ij}$.

Locality preserving projection. LPP is a linear approximation of the nonlinear Laplacian Eigenmap. The algorithmic procedure starts with constructing the adjacency graph from feature matrix $\boldsymbol{F}$. The item similarity matrix $\boldsymbol{S}$ learned from (1) can be used for this task. Then, we need to solve the generalized eigenvector problem:

$$\boldsymbol{F}^T\boldsymbol{LFw} = \gamma\boldsymbol{F}^T\boldsymbol{DFw}, \tag{2}$$

where $\boldsymbol{D}$ is a diagonal matrix, of which the i-th diagonal entry equals $\sum_j^n \frac{s_{ij}+s_{ji}}{2}$; $\boldsymbol{L}$ is a Laplacian matrix for $\boldsymbol{S}$, i.e., $\boldsymbol{L} = \boldsymbol{D} - \frac{\boldsymbol{S}+\boldsymbol{S}^T}{2}$. The projection matrix $\boldsymbol{W}$ is formed as $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k)$, where eigenvector $\boldsymbol{w}_i$ corresponds to eigenvalue γ_i, which is in an ascending order as $\gamma_1 \le \cdots \le \gamma_d$. The linear combination $\boldsymbol{FW}$ denotes the projection of side information in a low-dimensional space.

The proposed model. Putting (1) and (2) together forms our proposed optimization problem as follows:

$$\min_{\boldsymbol{S},\boldsymbol{W}} \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{RS}\|_F^2 + \frac{\alpha}{2}\sum_{i,j}^{n}\left(\|\boldsymbol{W}^T\boldsymbol{f}_i - \boldsymbol{W}^T\boldsymbol{f}_j\|_2^2 s_{ij}\right) + \frac{\lambda}{2}\|\boldsymbol{S}\|_F^2, \tag{3}$$

such that $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$; $\boldsymbol{s}_j^T\boldsymbol{1} = 1, \forall j = 1, \ldots, n$; $0 \le s_{ij} \le 1$, $\forall i, j = 1, \ldots, n$; and $s_{jj} = 0, \forall j = 1, \cdots, n$. Rather than impose the constraint $\boldsymbol{W}^T\boldsymbol{F}^T\boldsymbol{DFW} = \boldsymbol{I}$ on $\boldsymbol{W}$ according to LPP, we directly assume $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$ to learn a distinctive feature space. Besides, we regularize s_{ij} by $\|\boldsymbol{W}^T\boldsymbol{f}_i - \boldsymbol{W}^T\boldsymbol{f}_j\|_2^2$ instead of $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2$ for two reasons. First, the model is formulated as a joint learning optimization problem so as to achieve dimensionality reduction and top-N recommendation simultaneously. We will show later in Section 3.3 that optimizing $\boldsymbol{W}$ is under the framework of LPP. Second, the training of the item similarity matrix $\boldsymbol{S}$ is enhanced in the projected low-dimensional feature space. We argue that incorporating LPP is able to not only preserve locality but also improve item similarity, which is explained below.

Denote the projection matrix as $\boldsymbol{W} = (\boldsymbol{p}_1^T, \ldots, \boldsymbol{p}_d^T)^T$, where $\boldsymbol{p}_i$ is a k-dimensional row vector, representing the embedding of feature i. Though projection, each feature is represented by k distinctive aspects. We contend that the "synonyms" (different but semantically similar features) will have closer embeddings through LPP under the assumption that the synonyms are likely to appear in items with high similarities. Therefore, items containing synonyms will get closer in the projected space, which can further guide the learning of similarity towards more similar.

Once the solution $(\boldsymbol{W}^*, \boldsymbol{S}^*)$ are obtained, we can recover the item-user recommendation score matrix $\tilde{\boldsymbol{R}}$ by setting $\tilde{\boldsymbol{R}} = \boldsymbol{RS}^*$. We then rank the scores for unrated items of each user in a non-increasing order and recommend the first N items.

3.3 Solution

The optimization problem defined above is non-convex in terms of $\boldsymbol{S}, \boldsymbol{W}$ together. Thus, it is unrealistic to expect an algorithm to find the global minimum. In what follows, we derive an alternative iterative algorithm to solve the problem.

Fix $\boldsymbol{W}$ *update* $\boldsymbol{S}$. We first define the Lagrange function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{s}_j, \boldsymbol{\varphi}_j, \theta_j, \xi_j) =& \frac{1}{2}\|\boldsymbol{r}_j - \boldsymbol{R}\boldsymbol{s}_j\|_2^2 + \frac{\alpha}{2}\boldsymbol{q}_j^T\boldsymbol{s}_j + \theta_j\boldsymbol{s}_j^T\boldsymbol{1} + \\ & \frac{\lambda}{2}\boldsymbol{s}_j^T\boldsymbol{s}_j + \boldsymbol{\varphi}_j^T\boldsymbol{s}_j + \xi_j s_{jj},\end{aligned} \tag{4}$$

where $\boldsymbol{\varphi}_j, \theta_j, \xi_j$ $(j = 1, \ldots, n)$ are the lagrangian multipliers, $q_{ij} = \|\boldsymbol{W}^T\boldsymbol{f}_i - \boldsymbol{W}^T\boldsymbol{f}_j\|_2^2$ and $\boldsymbol{1}$ is the vector with all elements equal 1. The partial derivation of $\mathcal{L}$ w.r.t $\boldsymbol{s}_j$ is

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{s}_j} = \boldsymbol{R}^T\boldsymbol{R}\boldsymbol{s}_j - \boldsymbol{R}^T\boldsymbol{r}_j + \frac{\alpha}{2}\boldsymbol{q}_j + \theta_j\boldsymbol{1} + \lambda\boldsymbol{s}_j + \boldsymbol{\varphi}_j + \xi_j\boldsymbol{e}_j, \tag{5}$$

where $\boldsymbol{e}_j$ is the vector with only the j-th element equal 1 and others 0. A closed-form solution could be derived as follows:

$$\boldsymbol{s}_{ij} = \begin{cases}\left[\left(\boldsymbol{R}^T\boldsymbol{R} + \lambda\boldsymbol{I}\right)^{-1}\left(\boldsymbol{R}^T\boldsymbol{r}_j - \frac{\alpha}{2}\boldsymbol{q}_j - \theta_j\boldsymbol{1}\right)\right]_{i+}, & \text{if } i \neq j \\ 0, & \text{if } i = j,\end{cases} \tag{6}$$

where $\boldsymbol{R}^T\boldsymbol{R} + \lambda\boldsymbol{I}$ is positive definite if $\lambda > 0$ and $\theta_j = \boldsymbol{s}_j^T\boldsymbol{R}^T\boldsymbol{r}_j - \boldsymbol{s}_j^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{s}_j - \frac{\alpha}{2}\boldsymbol{s}_j^T\boldsymbol{q}_j - \lambda\boldsymbol{s}_j^T\boldsymbol{s}_j$; $[\cdot]_{i+}$ is the operator to take the i-th element of the vector if it is not less than 0, otherwise 0.

Fix $\boldsymbol{S}$ *update* $\boldsymbol{W}$. To update $\boldsymbol{W}$, we first introduce the following equation, which is based on the theory of spectral analysis:

$$\frac{1}{2}\sum_{i,j}\left(\|\boldsymbol{W}^T\boldsymbol{f}_i - \boldsymbol{W}^T\boldsymbol{f}_j\|_2^2 s_{ij}\right) = \mathrm{Tr}\left(\boldsymbol{W}^T\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}\boldsymbol{W}\right). \tag{7}$$

Hence, the problem is equivalent to solving

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} \mathrm{Tr}\left(\boldsymbol{W}^T\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}\boldsymbol{W}\right). \tag{8}$$

Applying the Karush-Kuhn-Tucker (KKT) first-order optimality conditions, we derive

$$\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}\boldsymbol{W} = \gamma\boldsymbol{W}, \tag{9}$$

and the solution is formed by the k eigenvectors of $\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}$ corresponding to the k smallest eigenvalues. Note that $\boldsymbol{W}$ is updated under the framework of LPP.

4 EXPERIMENT

4.1 Setup

To evaluate the performance of our method on the task of top-N recommendation with side information, we perform experiments on different real-world datasets, respectively, CUL[1], Enron[2] and Yahoo.[3] CUL is an online service that allows researchers to add scientific articles to their libraries. For each user, the articles added in his or her library are considered as preferred articles, from which titles and abstracts are collected and used as side information. Enron1 and Enron2 represent the two largest mailbox extracted from Enron Email. The data is composed of email messages released during investigation of the Federal Energy Regulatory Commission against the Enron Corporation. By regarding the email content as side information, we predict the most likely recipients of new messages. Yahoo contains a small sample of the Yahoo! Movies community's preferences for various movies, rated on a scale from A+ to F, binarized to 0 or 1. The dataset also contains a large amount of side information about many movies. The statistics of the datasets are summarized in Table 1.

Table 1: Statistics of datasets.

Dataset	#users	#items	#feeds	density	#features
Enron1	663	1,773	1,588	0.14%	25,133
Enron2	953	5,366	3,401	0.07%	32,063
Yahoo	7,594	8,641	19,434	0.03%	7,823
CLU	9,537	8,222	29,352	0.04%	6,860

To comprehensively understand the effectiveness of the methods, we adopt 5-time Leave-One-Out Cross Validation. The evaluation of the model is conducted by comparing the recommendation list of each user with the item of that user in the test set. The recommendation quality is measured using the Hit Rate (HR) and the Average Reciprocal Hit Rank (ARHR).[4] We evaluate the performance of our proposed method on top-N recommendation.

In this set of experiments, we refer to our method as Prism (Projection regularized item similarity model). To evaluate its performance, Prism is first compared with SLIM to demonstrate the need to utilize side information when feedback is sparse. The performance of CoSim (a pure content-based method [1]) is evaluated to show the quality of side information. To appreciate the effectiveness of dimensionality reduction, the performance of Baseline, formulated in Equation (1), is also evaluated. We also compare Prism with state-of-the-art top-N recommendation methods with side information, including SSLIM [6], UFSM [1] and the method proposed in [9] (referred to as PCF). Parameters of all methods are carefully tuned through grid search.

4.2 Results and analysis

We vary the size of recommendation list, and find that Prism always achieves the best results. Table 2 shows the result of comparisons over four datasets with top-10 items recommended. By looking at the results achieved by SLIM and CoSim, we can characterize the datasets. Overall speaking, SLIM performs inferiorly to CoSim on both Enron1 and Enron2, whereas the order is reversed on Yahoo and CUL. This shows that while all datasets are sparse with respect to user feedback information, the side information of Enron is of high quality and more relevant for recommendation. As the Enron datasets are of higher dimensionality, a significant performance gain is expected with Prism on Enron1 and Enron2. To verify this, we scrutinize the results of Prism and Baseline, and find that the improvement of Prism over Baseline is much more evident on Enron1 and Enron2 than that on Yahoo and CUL. These results demonstrate the effectiveness of incorporating LPP for recommendation involving high-dimensional side information.

As for the comparison with other methods, Prism achieves the best results over all tested datasets, especially on Enron2, which has the highest dimensionality of side information. The recommendation accuracy of Prism on this dataset enjoys a performance gain up to 21.2% on HR and 36.8% on ARHR over state-of-the-art methods. This demonstrates the effectiveness of Prism. On the Yahoo

[1]CiteULike: http://www.citeulike.org/

[2]Enron Mail Box: https://www.cs.cmu.edu/~enron/

[3]Yahoo! Movies: https://webscope.sandbox.yahoo.com/

[4]For each user, we recommend N items, where $N = 5, 10, 15, 20$. Due to space limitations, we only present the result with $N = 10$.

Table 2: Comparison of top-N recommendation algorithms.

Method	Enron1			Enron2		
	Parameters	HR10	ARHR10	Parameters	HR10	ARHR10
CoSim	—	0.1408	0.0992	—	0.1460	0.1196
SLIM	$\beta = 0.6$, $\lambda = 0.2$	0.0865	0.0347	$\beta = 0.1$, $\lambda = 0.5$	0.1569	0.0668
SSLIM1	$\alpha = 0.9$, $\beta = 0.1$, $\lambda = 0.2$	0.2032	0.0966	$\alpha = 0.8$, $\beta = 0.1$, $\lambda = 0.5$	0.2204	0.1119
SSLIM2	$\alpha = \beta = 0.1$, $\lambda = 0.2$	0.0853	0.0327	$\alpha = 0.1$, $\beta = 0.1$, $\lambda = 0.5$	0.1547	0.0733
UFSM$_{rmse}$	$l = 1.\lambda = 0.1$, $\mu_1 = 0.01$, $\mu_2 = 10^{-5}$	0.1485	0.1059	$l = 4$, $\lambda = 0.1$, $\mu_1 = 0.1$, $\mu_2 = 10^{-5}$	0.1693	0.1273
UFSM$_{bpr}$	$l = 1.\lambda = 10^{-5}$, $\mu_1 = 0.01$, $\mu_2 = 10^{-4}$	0.1416	0.1040	$l = 3$, $\lambda = 10^{-5}$, $\mu_1 = 0.01$, $\mu_2 = 0.01$	0.1511	0.1142
PCF	$\beta = 0.5$, $\gamma = 10.0$, $\lambda = 500$	0.2013	0.1011	$\beta = 0.2$, $\gamma = 5$, $\lambda = 1000$	0.2318	0.1201
Baseline	$\alpha = 1.0$, $\lambda = 0.3$	0.0966	0.0435	$\alpha = 0.9$, $\lambda = 0.5$	0.1679	0.0850
Prism	$k = 100$, $\alpha = 2.0$, $\lambda = 0.2$	**0.2153**	**0.1091**	$k = 100$, $\alpha = 0.3$, $\lambda = 0.4$	**0.2810**	**0.1742**

Method	Yahoo			CUL		
	Parameters	HR10	ARHR10	Parameters	HR10	ARHR10
CoSim	—	0.0241	0.0106	—	0.1238	0.0559
SLIM	$\beta = 0.9$, $\lambda = 0.5$	0.0558	0.0181	$\beta = 1.0$, $\lambda = 0.5$	0.1961	0.0758
SSLIM1	$\alpha = 0.7$, $\beta = 0.5$, $\lambda = 0.1$	0.0543	0.0193	$\alpha = 0.9$, $\beta = 1.0$, $\lambda = 0.5$	0.1916	0.0733
SSLIM2	$\alpha = \beta = 0.1$, $\lambda = 0.5$	0.0485	0.0181	$\alpha = 0.1$, $\beta = 0.1$, $\lambda = 0.5$	0.2223	0.0873
UFSM$_{rmse}$	$l = 6$, $\lambda = 0.1 = \mu_1 = 0.1$, $\mu_2 = 10^{-4}$	0.0408	0.0195	$l = 5$, $\lambda = 10^{-5}$, $\mu_1 = 10^{-4}$, $\mu_2 = 10^{-5}$	0.1821	0.0705
UFSM$_{bpr}$	$l = 5$, $\lambda = \mu_1 = \mu_2 = 1^{-5}$	0.0400	0.0192	$l = 5$, $\lambda = 10^{-5}$, $\mu_1 = 0.01$, $\mu_2 = 10^{-5}$	0.1942	0.0803
PCF	$\beta = 1.0$, $\gamma = 10$, $\lambda = 2000$	0.0556	0.0208	$\beta = 0.8$, $\gamma = 5$, $\lambda = 2000$	0.2167	0.0834
Baseline	$\alpha = 1.0$, $\lambda = 0.5$	0.0618	0.0230	$\alpha = 1.0$, $\lambda = 0.1$	0.2118	0.0864
Prism	$k = 300$, $\alpha = 0.9$, $\lambda = 0.1$	**0.0672**	**0.0232**	$k = 200$, $\alpha = 0.3$, $\lambda = 0.1$	**0.2247**	**0.0936**

dataset SSLIM and UFSM actually degrade the accuracy compared with SLIM. While PCF increases it, the increment is limited. This should be attributed to the poor quality of side information. By contrast, Prism improves it, exhibiting the robustness of Prism; that is, even on the dataset where side information is of limited correlation to recommendation, the preferable result could be expected. This robustness is also displayed on CUL, which takes good user feedback but poor side information. It seems that CUL is more suitable to the methods that loosely couple with side information like SSLIM2. In this case, Prism is still able to achieve quite competitive performance, as the relevance of side information is improved through dimensionality reduction and α is tuned small to emphasize more on feedback information. The performance on Enron1 is not that distinctive. As the side information is of high quality, the methods that tightly couple with side information stand out (SSLIM1 and PCF). On the other hand, as the feature dimensionality is lower than that on Enron2, dimensionality reduction is not equally effective.

5 CONCLUSION

In this paper, we showed the problems encountered when utilizing high-dimensional side information to enhance the performance of recommendation, which had not been well investigated by existing literature. We proposed a novel method to address the challenge, namely Projection regularized item similarity model–Prism. The method integrates LPP and top-n recommendation into a joint learning algorithm. Under the novel framework, LPP not only resolved the issue brought by high dimensionality, but also improved the relevance of item similarity. We conducted extensive experiments and the results demonstrated the superiority of Prism.

Acknowledgment. This work was partially supported by NSFC No. 61402494, 61402498, 71690233, NSF Hunan No. 2015JJ4009, Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] A. Elbadrawy and G. Karypis. User-specific feature-based similarity models for top-n recommendation of new items. *ACM Trans. Intel. Syst. Tech.*, 6(3):33, 2015.

[2] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *10th IEEE International Conference on Data Mining, ICDM 2010, Sydney, Australia*, pages 176–185, 2010.

[3] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16: Annual Conference on Neural Information Processing Systems 2003, NIPS 2003, Vancouver and Whistler, British Columbia, Canada*, pages 153–160, 2003.

[4] J. Lu, G. Liang, J. Sun, and J. Bi. A sparse interactive model for matrix completion with side information. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, NIPS 2016, Barcelona, Spain*, pages 4071–4079, 2016.

[5] X. Ning and G. Karypis. SLIM: sparse linear methods for top-n recommender systems. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada*, pages 497–506, 2011.

[6] X. Ning and G. Karypis. Sparse linear methods with side information for top-n recommendations. In *Sixth ACM Conference on Recommender Systems, RecSys 2012, Dublin, Ireland, September 9-13, 2012*, pages 155–162, 2012.

[7] F. Ricci, L. Rokach, and B. Shapira, editors. *Recommender Systems Handbook.* Springer, 2015.

[8] M. Saveski and A. Mantrach. Item cold-start recommendations: learning local collective embeddings. In *Eighth ACM Conference on Recommender Systems, RecSys 2014, Foster City, Silicon Valley, CA, USA*, pages 89–96, 2014.

[9] F. Zhao, M. Xiao, and Y. Guo. Predictive collaborative filtering with side information. In *Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA*, pages 2385–2391, 2016.