Leveraging High-Dimensional Side Information for Top-N Recommendation

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

Xiang Zhao
National University of Defense Technology
xiangzhao@nudt.edu.cn

Yang Wang
The University of New South Wales
wangy@cse.unsw.edu.au

Philip S. Yu
University of Illinois at Chicago
psyu@uic.edu

ABSTRACT

Top-N recommender systems typically utilize side information to address the problem of data sparsity. As nowadays side information is growing towards high dimensionality, the performances of existing methods deteriorate in terms of both effectiveness and efficiency, which imposes a severe technical challenge. In order to take advantage of high-dimensional side information, we propose in this paper an embedded feature selection method to facilitate top-N recommendation. In particular, we propose to learn feature weights of side information, where zero-valued features are naturally filtered out. We also introduce non-negativity and sparsity to the feature weights, to facilitate feature selection and encourage low-rank structure. Two optimization problems are accordingly put forward, respectively, where the feature selection is tightly or loosely coupled with the learning procedure. Augmented Lagrange Multiplier and Alternating Direction Method are applied to efficiently solve the problems. Experiment results demonstrate the superior recommendation quality of the proposed algorithm to that of the state-of-the-art alternatives.

Keywords

Top-N Recommendation; High Dimensionality; Side Information; Feature Selection

1. INTRODUCTION

Recommender systems typically make use of user feedback, e.g., purchases, ratings, reviews, clicks, and check-ins, to produce the recommendations. Most of the methods can be put broadly into two categories - latent space methods and neighborhood-based methods. Existing research [11, 18, 22] suggests that latent space methods are superior for solving the *rating prediction* problem, while neighborhood methods are shown to be better for the *top-N recommendation* problem. We focus ourselves on the latter in this paper.

Top-N recommender systems are widely adopted by the majority of e-commerce websites 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, *item*-based methods stand out, including item-based k-nearest-neighbor (itemkNN) [11] and sparse linear methods (SLIM) [22], which are shown to outperform *user*-based schemes [10].

While a number of algorithms have been developed, the *data sparsity* issues faced across different recommender systems still remain open, i.e., the feedback for some users or items is little or even entirely missing. With the increasing availability of the additional information associated with items, referred to as *side information*[1], recent literature has developed many *hybrid* algorithms to relieve the aforementioned problem [1, 13, 23, 26], since side information is often relevant to recommendation.

Nonetheless, in the era of big data, side information is born with *high dimensionality*. For example, side information is usually the text descriptions about items [28, 29], and when regarding each unique term in the corpus as one dimension of feature, it is evidently high-dimensional. Side information can also be in the forms of images [16] or even videos [24], in which cases the dimensionality is even higher. However, existing methods using side information consider little about the heavy impact of high-dimensional side information on accuracy and efficiency. Specifically, it is observed that many features in side information are useless or not relevant to recommendation, and thus, including these features in the algorithms would harm the effectiveness of recommendation. An immediate remedy is to employ feature selection to escape the curse of dimensionality, which reduces the dimensionality by selecting a subset of features from the input feature set. In general, feature selection methods can be categorized into three families - filter-based, wrapper-based and embedded methods [15]. Among them, embedded methods are demonstrated to be superior in many aspects,

[1]To name a few, the descriptions of movies in movie recommendation, the applicant's resumes in job matching, the content of emails in spam detection, the reviews of items in online shopping, and so forth.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

and receive more attention [21]. It is believed that feature quality is usually domain-specific, i.e., for different scenarios the importance of features is accordingly different, where embedded feature selection becomes more preferable. As a consequence, we adopt embedded methods in this research.

In this paper, we propose a novel embedded feature selection method for top-N recommendation to harness high-dimensional side information. We propose two optimization problems to learn the weights of features of side information. To avoid the impact from the high dimensionality and make the recommendation fast, we exploit embedded feature selection, where the goals of learning and feature selection are *jointly* and *iteratively* achieved. Specifically, the proposed method learns a sparse feature weights vector for each individual feature of side information, and the features with zero values are naturally filtered out. This is accomplished by solving the optimization problems, with sparsity and non-negativity enforced for the feature weights. Experimental evaluation shows that the proposed method enjoys a performance gain of 25% at most.

In summary, we make the following contributions:

- We investigate the problem of top-N recommendation leveraging high-dimensional side information, which has not been well explored by existing literature. To the best of our knowledge, this is among the first attempts to look into the issues brought by side information for top-N recommendation in the big data era;
- To alleviate the impact of high dimensionality of side information, we propose an embedded feature selection method to learn the feature weights, where the tasks of feature selection and learning are jointly achieved and mutually enhanced. As far as we understand, this is also might be the first time that feature selection closely meets recommendation;
- We conduct extensive experiments to appreciate the effectiveness of the proposed method. We experiment on real-life datasets from two different application domains, and the results confirm that the proposed method well resolves the issue, and outperforms other state-of-the-art algorithms.

The remainder of this paper is organized as follows: Section 2 introduces the notations used in this paper, followed by the discussion of related work in Section 3. We formulate the model in Section 4, and the proposed optimization problems are solved in Section 5. We conduct extensive experiments on real-life datasets, and present the results in Section 6. Section 7 draws the conclusions of the paper.

2. NOTATIONS

We introduce the relative notations in this section. All vectors and matrices are represented by bold letters, where lower case letters for vectors (e.g. $\boldsymbol{x}$) and upper case letters for matrices (e.g. $\boldsymbol{X}$), and all constant parameters are represented by Greek letters (e.g. α). Given matrix $\boldsymbol{X}$, x_{ij} represents the entry at i^{th} row and j^{th} column. Given vector $\boldsymbol{x}$, x_i represents the entry of i^{th} element. $\|\cdot\|_1$ indicates the ℓ_1-norm, e.g., $\|\boldsymbol{X}\|_1 = \sum_i \sum_j |x_{ij}|$, $\|\boldsymbol{x}\|_1 = \sum_i |x_i|$. $\|\cdot\|_F$ indicates the Frobenius norm, e.g., $\|\boldsymbol{X}\|_F = \left(\sum_i \sum_j x_{ij}^2\right)^{1/2}$. And $\|\cdot\|_2$ indicates the ℓ_2-norm for vector, e.g., $\|\boldsymbol{x}\|_2 = \left(\sum_i x_i^2\right)^{1/2}$. We define $\circ$ as the element-wise product between two vectors: $\boldsymbol{x} \circ \boldsymbol{y} = (x_1 y_1, \cdots, x_n y_n)$.

We denote m, n, t for the number of users, items and features respectively. Let $U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{v_1, v_2, \ldots, v_n\}$ be the sets of all users and all items, respectively. The feedback (both explicit and implicit) shows the items the users have purchased, viewed or rated, which is denoted by $\boldsymbol{R} \in \mathbb{R}^{m \times n}$. We treat the feedback into binary value, that is, if user u provided feedback for item i, r_{ui} (the entry of $\boldsymbol{R}$) 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 $[0, 1]$. We further introduce the feature matrix (side information), denoted by $\boldsymbol{F} \in \mathbb{R}^{n \times t}$. The feature weights are represented by $\boldsymbol{w} \in \mathbb{R}^t$, where each entry w_k represents the global feature weight for the k^{th} feature. We further denote $\boldsymbol{d}_i \in \mathbb{R}^n$ for the weighted feature vector for item i, that is, $\boldsymbol{d}_i = \boldsymbol{f}_i \circ \boldsymbol{w}^{1/2}$ and thus $\boldsymbol{d}_i \cdot \boldsymbol{d}_j = \left(\boldsymbol{f}_i \circ \boldsymbol{f}_j\right) \cdot \boldsymbol{w}$. We also denote $\boldsymbol{p}_k \in \mathbb{R}^t$ as the normalized weight vector for feature k, that is, $p_{ik} = f_{ik} / \|\boldsymbol{d}_i\|_2$.

3. RELATED WORK

In this section, we retrospect the research following SLIM, the work related to the hybrid methods utilizing side information and feature selection approaches, and present them in the following subsections, respectively.

3.1 Top-N Recommendation

Recently, a novel top-N recommendation method - sparse linear method (SLIM) was proposed [22], which generates recommendation lists by learning a sparse similarity matrix. SLIM solves the following regularized optimization problem:

$$\begin{aligned} \min_{\boldsymbol{S}} \mathcal{O} = \quad & \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{S}\|_F^2 + \lambda\|\boldsymbol{S}\|_1 \\ s.t. \quad & \operatorname{diag}(\boldsymbol{S}) = \boldsymbol{0}, \\ & \boldsymbol{S} \geq 0, \end{aligned}$$

where the constants β and λ are regularization parameters. The non-negativity constraint is used such that the vector estimated contains positive coefficients. The $\operatorname{diag}(\boldsymbol{S}) = \boldsymbol{0}$ is imposed to avoid the trivial results.

However, an inherent limitation of SLIM is that it can only model relations between items that have been co-purchased or co-rated by at least some users. Follow-on research attempted to address the problem from different aspects, factorizing the similarity matrix [18, 31] and introducing low-rank approximation [8, 19], respectively. Related work following this line of research also includes [9, 12, 33]. Lately, a new model based on SLIM was proposed to capture the difference in the preferences between different user subsets [10].

3.2 Hybrid Recommendation

To address the data sparsity problem, many hybrid methods were proposed, utilizing side information. SSLIM [23] proposes to extend SLIM into a joint learning model. LCE [25] collectively decomposes the content and the collaborative matrices in a common low-dimensional space. [2, 3] decouple the recommendation into completion and transduction, and provide provable guarantee. The method proposed by [14] leverages both user activities and content matching between user and item profiles to optimize the global term weights. In the aforementioned work, the problem brought by high di-

mensionality of side information is not noted or investigated, and hence, their performances could be heavily affected by high-dimensional side information.

3.3 Feature Selection

Various methods of feature selection have been proposed, and can be classified into three distinct types, i.e., filter method [17, 32], wrapper method [27] and embedded method [6, 21, 30]. The filter model evaluates features without involving any learning algorithm. The wrapper model requires a learning algorithm and uses its performance to evaluate the goodness of features. Algorithms of the embedded model incorporate feature selection as part of the learning process, and use the objective function of the learning model to guide searching for relevant features. For various types of feature selection methods, readers may refer to [7] for a recent survey.

4. MODEL DESCRIPTION

In this section, we propose an embedded feature selection method on the top of SLIM, named as HSLIM, which is short for High-Dimensional Side Information Utilized Sparse LInear Methods. HSLIM is specifically designed to utilize high-dimensional side information.

4.1 Side Information

We assume the availability of profiles associated with items, e.g., the side information. In this paper, we assume the side information is in the form of text as it is often the case, such as the description of movies, the content of documents, the review of items and so on. However, we may readily extend our proposed method to other forms of side information. Thus the dimension of side information is then regarded as each individual term in corpus.

We then introduce feature weight for each individual dimension of side information to measure its importance, and the features are selected accordingly. Intuitively, the feature weights could be measured by the *Inverted Document Frequency* (IDF) used in text similarity. However, it has been shown by [14] that IDF is not optimal or even reliable in recommendation. Thus, we optimize feature weights by using feedback as training data, and hence, the domain-specific feature weights are generated.

Given the side information matrix $\boldsymbol{F} \in \mathbb{R}^{n\times t}$, where each entry f_{ik} represents the number of occurrence of term k in the side information of item i, and the feature weights $\boldsymbol{w}$, we apply cosine function to measure the similarity between items. The similarity between item i and j then is

$$\begin{aligned}\cos\left(\boldsymbol{d}_i, \boldsymbol{d}_j\right) &= \frac{\boldsymbol{d}_i \cdot \boldsymbol{d}_j}{\|\boldsymbol{d}_i\|_2 \|\boldsymbol{d}_j\|_2} \\ &= \frac{\sum_{k=1}^{t} f_{ik} f_{jk} w_k}{\left[\sum_{k=1}^{t} f_{ik}^2 w_k\right]^{\frac{1}{2}} \left[\sum_{k=1}^{t} f_{jk}^2 w_k\right]^{\frac{1}{2}}}.\end{aligned}$$

4.2 Optimization Problem

Denote as $\boldsymbol{S}^w$ the item similarities derived from side information, the entries of which are computed according to Equation (1). We force $s_{ii}^w = 0, i = 1, 2, \ldots, n$ to avoid trivial answers.

$$s_{ij}^w = \begin{cases} \cos\left(\boldsymbol{d}_i, \boldsymbol{d}_j\right), & \text{if } i \neq j; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

We incorporate $\boldsymbol{S}^w$ into SLIM model so as to jointly learn $\boldsymbol{w}$ and the sparse coefficient matrix $\boldsymbol{S}$. Different in how to utilize $\boldsymbol{S}^w$, we have proposed two optimization problems.

HSLIMtf (HSLIM tightly coupled with feature selection) directly substitutes the sparse coefficient matrix $\boldsymbol{S}$ by $\boldsymbol{S}^w$. We formulate the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{w}} \mathcal{O} = \quad & \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}^w\|_F^2 + \lambda_1 \|\boldsymbol{S}^w\|_1 + \lambda_2 \|\boldsymbol{w}\|_1 \\ s.t. \quad & \boldsymbol{w} \geq \boldsymbol{0},\end{aligned}$$

where we penalize ℓ_1-norm on $\boldsymbol{S}^w$ to encourage sparsity, as suggested by SLIM. We also penalize ℓ_1-norm on $\boldsymbol{w}$ for that we select feature in the light of feature weights $\boldsymbol{w}$, where the zero valued dimensions are dropped. We also impose non-negativity on $\boldsymbol{w}$ as $\boldsymbol{d}_i = \boldsymbol{f}_i \circ \boldsymbol{w}^{1/2}$ and $\boldsymbol{w}$ is supposed to be in the domain of real number. According to Equation (1), the non-negativity ($\boldsymbol{S}^w \geq \boldsymbol{0}$) and non-triviality ($\text{diag}(\boldsymbol{S}^w) = \boldsymbol{0}$) naturally hold. Thus, the constraints are omitted here.

HSLIMlf (HSLIM loosely coupled with feature selection) penalizes the difference between $\boldsymbol{S}$ and $\boldsymbol{S}^w$, rather than directly substitutes $\boldsymbol{S}$. The problem is then formulated as:

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{S}} \mathcal{O} = \quad & \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{S} - \boldsymbol{S}^w\|_F^2 + \\ & \lambda_1 \|\boldsymbol{S}\|_1 + \lambda_2 \|\boldsymbol{w}\|_1 \\ s.t. \quad & \text{diag}(\boldsymbol{S}) = 0, \\ & \boldsymbol{w} \geq \boldsymbol{0}, \\ & \boldsymbol{S} \geq \boldsymbol{0},\end{aligned}$$

where $\frac{\alpha}{2}\|\boldsymbol{S} - \boldsymbol{S}^w\|_F^2$ regularizes the sparse matrix $\boldsymbol{S}$ and influences the learning of $\boldsymbol{w}$. α is introduced to balance the importance of side information.

HSLIMtf assumes the side information is rich and highly correlated with feedback, thus the substitution of $\boldsymbol{S}^w$ will mostly enhance the recommendation quality. Otherwise, HSLIMlf should be applied as it actually softens the requirement on $\boldsymbol{S}$. The selection of approaches in fact depends on the quality of side information. In real application, we start with HSLIMlf and conduct parameter selection for α. When it achieves good performance if $\alpha = 1$, we will switch to HSLIMtf as it suggests good quality of side information, and thus, HSLIMtf would achieve better results.

Once the optimization problem is solved, we can conduct matrix completion for matrix $\boldsymbol{R}$ for recommendation. The predicted user-item matrix, denoted by $\tilde{\boldsymbol{R}}$, is calculated as $\tilde{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{S}^w$ for HSLIMtf and $\tilde{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{S}$ for HSLIMlf. In addition, for item cold-start scenario, we calculate the similarities of the new item with other items through item profiles. As the incorporated feature selection procedure largely reduces the number of features in profile, the similarity calculation will be much faster, which consequently improves the efficiency for recommendation.

Note that we have introduced sparsity to feature weights $\boldsymbol{w}$ for feature selection. Naturally, the sparsity of $\boldsymbol{w}$ also encourages the low-rank structure of item similarity matrix, which is crucial to the performance of recommendation [8, 19]. In essence, the low-rank assumption is moti-

vated by the factor model. It is assumed that a few latent variables explain items' features, where the item feature is represented by $\tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{F}}$ is low-rank. In our proposed model, the weighted item feature matrix is represented as $\tilde{\boldsymbol{F}} = \boldsymbol{P}\text{diag}(\sqrt{\boldsymbol{w}})$, where $\boldsymbol{P}$ is the normalized feature matrix ($p_{ik} = f_{ik}/\|\boldsymbol{d}_i\|_2$). $\text{diag}(\sqrt{\boldsymbol{w}})$ is a diagonal matrix whose diagonal equals to $\sqrt{\boldsymbol{w}}$ and $\sqrt{\cdot}$ is element-wise square root. As penalizing sparsity on $\boldsymbol{w}$ increases the number of zeros in $\boldsymbol{w}$, and given that the rank of $\tilde{\boldsymbol{F}}$ equals to the number of non-zero values of $\boldsymbol{w}$, the learned feature matrix will be low-rank.

5. SOLUTION

In this section, we introduce how to solve the optimization problems. We will discuss how the model HSLIMtf is solved, and then the solution for HSLIMlf could be derived similarly (omitted in the interest of space).

The proposed optimization problem is difficult to optimize for the following three reasons: (1) We have forced the non-negativity constraint on feature weights $\boldsymbol{w}$; (2) We penalize ℓ_1-norms on $\boldsymbol{S}$ and $\boldsymbol{w}$, and the ℓ_1-norm is non-differentiable; and (3) The entries of similarity matrix $\boldsymbol{S}^w$ is not independent, as they are all related to $\boldsymbol{w}$. Thus it is also difficult to derive the derivative.

Subsequently, we first introduce auxiliary variables to decouple the objective function, and solve the following equivalent problem:

$$\begin{aligned} \min_{\boldsymbol{w}} \mathcal{O} = \quad & \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}_1\|_F^2 + \lambda_1\|\boldsymbol{S}_2\|_1 + \lambda_2\|\boldsymbol{w}\|_1 \\ s.t. \quad & \boldsymbol{S}_1 = \boldsymbol{S}_2 = \boldsymbol{S}^w, \\ & \boldsymbol{w} \geq 0. \end{aligned}$$

This can be solved by using Augmented Lagrange Multiplier (ALM) [5]. We resort to minimizing the following augmented Lagrangian function:

$$\begin{aligned} & L(\boldsymbol{w}, \boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{V}_1, \boldsymbol{V}_2) = \\ & \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}_1\|_F^2 + \lambda_1\|\boldsymbol{S}_2\|_1 + \lambda_2\|\boldsymbol{w}\|_1 \\ & + \frac{\beta}{2}\|\boldsymbol{S}_1 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_1}{\beta}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{S}_2 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_2}{\beta}\|_F^2, \end{aligned}$$

where $\beta > 0$ is the penalty parameter and $\boldsymbol{V}_1, \boldsymbol{V}_2$ are the Lagrange multipliers. We then apply Alternating Direction Method (ADM) to solve the problem.

$$\begin{aligned} & \boldsymbol{w}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{w} \geq \boldsymbol{0}} L(\boldsymbol{w}, \boldsymbol{S}_1^{(k)}, \boldsymbol{S}_2^{(k)}, \boldsymbol{V}_1^{(k)}, \boldsymbol{V}_2^{(k)}); \\ & \boldsymbol{S}_1^{(k+1)} \leftarrow \arg\min_{\boldsymbol{S}_1} L(\boldsymbol{w}^{(k+1)}, \boldsymbol{S}_1, \boldsymbol{S}_2^{(k)}, \boldsymbol{V}_1^{(k)}, \boldsymbol{V}_2^{(k)}); \\ & \boldsymbol{S}_2^{(k+1)} \leftarrow \arg\min_{\boldsymbol{S}_2} L(\boldsymbol{w}^{(k+1)}, \boldsymbol{S}_1^{(k+1)}, \boldsymbol{S}_2, \boldsymbol{V}_1^{(k)}, \boldsymbol{V}_2^{(k)}); \\ & \boldsymbol{V}_1^{(k+1)} \leftarrow \boldsymbol{V}_1^{(k)} - \gamma\beta(\boldsymbol{S}_1^{(k+1)} - \boldsymbol{S}_w^{(k+1)}); \\ & \boldsymbol{V}_2^{(k+1)} \leftarrow \boldsymbol{V}_2^{(k)} - \gamma\beta(\boldsymbol{S}_2^{(k+1)} - \boldsymbol{S}_w^{(k+1)}). \end{aligned}$$

Next, we are to investigate the updating rule for $\boldsymbol{w}, \boldsymbol{S}_1$ and $\boldsymbol{S}_2$.

5.1 Fix $\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{V}_1, \boldsymbol{V}_2$ Update $\boldsymbol{w}$

To update $\boldsymbol{w}$, we have the following problem:

$$\min_{\boldsymbol{w} \geq 0} \frac{\beta}{2}\|\boldsymbol{S}_1 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_1}{\beta}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{S}_2 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_2}{\beta}\|_F^2 + \lambda_2\|\boldsymbol{w}\|_1.$$

Then, based on the chain rule the partial derivative over w_k is:

$$\frac{\partial \mathcal{O}}{\partial w_k} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial \mathcal{O}}{\partial s_{ij}^w}\frac{\partial s_{ij}^w}{\partial w_k} + \lambda_2,$$

and the derivative of s_{ij}^w over w_k is:

$$\begin{aligned} \frac{\partial s_{ij}^w}{\partial w_k} &= \frac{1}{\|\boldsymbol{d}_i\|_2^2\|\boldsymbol{d}_j\|_2^2}\{f_{ik}f_{jk}\|\boldsymbol{d}_i\|_2\|\boldsymbol{d}_j\|_2 - \\ &\quad \left[\frac{\|\boldsymbol{d}_j\|_2}{2\|\boldsymbol{d}_i\|_2}f_{ik}^2 + \frac{\|\boldsymbol{d}_i\|_2}{2\|\boldsymbol{d}_j\|_2}f_{jk}^2\right]\boldsymbol{d}_i \cdot \boldsymbol{d}_j\Big\} \\ &= \frac{f_{ik}f_{jk}}{\|\boldsymbol{d}_i\|_2\|\boldsymbol{d}_j\|_2} - \frac{s_{ij}^w}{2}\left[\frac{f_{ik}^2}{\|\boldsymbol{d}_i\|_2^2} + \frac{f_{jk}^2}{\|\boldsymbol{d}_j\|_2^2}\right] \\ &= p_{ik}p_{jk} - \frac{s_{ij}^w}{2}(p_{ik}^2 + p_{jk}^2), \end{aligned}$$

where $p_{ik} = f_{ik}/\|\boldsymbol{d}_i\|_2$. Denote o_{ij} for $\frac{\partial \mathcal{O}}{\partial s_{ij}^w}$, we can continue to deduce as follows:

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial w_k} &= \sum_{i=1}^{n}\sum_{j=1}^{n} o_{ij}\left\{p_{ik}p_{jk} - \frac{s_{ij}^w}{2}\left[(p_{ik})^2 + (p_{jk})^2\right]\right\} + \lambda_2 w_k \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ik}o_{ij}p_{jk} - \sum_{i=1}^{n}\sum_{j=1}^{n} o_{ij}s_{ij}^w(p_{ik})^2 + \lambda_2. \end{aligned}$$

By further defining $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ as the matrix of o_{ij} and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ as the diagonal matrix with the i^{th} element of principal diagonal equals $\sum_{j=1}^{n} o_{ij}s_{ij}^w$, we have:

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial w_k} &= (\boldsymbol{p}_k)^T(\boldsymbol{Q} - \boldsymbol{D})\boldsymbol{p}_k + \lambda_2 \\ &= (\boldsymbol{p}_k)^T\boldsymbol{L}\boldsymbol{p}_k + \lambda_2, \end{aligned} \tag{2}$$

where $\boldsymbol{L} = \boldsymbol{Q} - \boldsymbol{D}$. We proceed to discuss how the matrix $\boldsymbol{Q}$ is obtained. Specifically,

$$\begin{aligned} & \boldsymbol{Q} = \frac{\partial \mathcal{O}}{\partial \boldsymbol{S}_w} = \\ & -\beta\left(\boldsymbol{S}_1 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_1}{\beta}\right) - \beta\left(\boldsymbol{S}_2 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_2}{\beta}\right). \end{aligned} \tag{3}$$

Due to the non-negativity of w_k, we employ the projected gradient methods to solve it [20]. The following rule is defined for updating:

$$\boldsymbol{w}^{(t+1)} = P\left[\boldsymbol{w}^{(t)} - \alpha^{(t)}\frac{\partial \mathcal{O}}{\partial \boldsymbol{w}^{(t)}}\right],$$

where

$$P[x] = \begin{cases} x, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

During each iteration, to ensure sufficient decrease, the following inequality needs to be conformed:

$$\mathcal{O}(\boldsymbol{w}^{k+1}) - \mathcal{O}(\boldsymbol{w}^k) \leq \sigma(\boldsymbol{w}^{k+1} - \boldsymbol{w}^k)^T\frac{\partial \mathcal{O}}{\partial \boldsymbol{w}^k},$$

where σ is empirically set as 0.01.

5.2 Fix $\boldsymbol{w}, \boldsymbol{V}_1, \boldsymbol{V}_2$ Update $\boldsymbol{S}_1, \boldsymbol{S}_2$

To update $\boldsymbol{S}_1$, we have:

$$\min_{\boldsymbol{S}_1} \frac{1}{2}\|\boldsymbol{R} - \boldsymbol{R}\boldsymbol{S}_1\|_F^2 + \frac{\beta}{2}\|\boldsymbol{S}_1 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_2}{\beta}\|_F^2,$$

and the analytical solution could be derived as follows:

$$\boldsymbol{S}_1 = (\boldsymbol{R}^T\boldsymbol{R} + \beta\boldsymbol{I})^{-1}(\boldsymbol{R}^T\boldsymbol{R} + \beta\boldsymbol{S}^w - \boldsymbol{V}_1). \quad (4)$$

To update $\boldsymbol{S}_2$, we have:

$$\min_{\boldsymbol{S}_2} \lambda_1\|\boldsymbol{S}_2\|_1 + \frac{\beta}{2}\|\boldsymbol{S}_2 - \boldsymbol{S}^w + \frac{\boldsymbol{V}_2}{\beta}\|_F^2.$$

Prior to the discussion, we first introduce Lemma 1 [4].

LEMMA 1. *For $\lambda > 0$, the solution of the problem*

$$\min_{\mathbf{X}} \lambda\|\mathbf{X}\|_1 + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2$$

is given by $\mathbf{X}^* = \text{shrink}_\lambda(\mathbf{Y})$, where

$$\text{shrink}_\lambda(x) = \max\{|x| - \lambda, 0\} \cdot \text{sgn}(x).$$

and the operator shrink *is applied element wise on $\mathbf{Y}$.*

Therefore, the solution of the problem is given by

$$\boldsymbol{S}_2 = \text{shrink}_{\lambda_1}(\boldsymbol{S}^w - \frac{\boldsymbol{V}_2}{\beta}). \quad (5)$$

We finally formulate the complete training for HSLIMtf into Algorithm 1.

Algorithm 1: Training Procedure for HSLIMtf

Input : Feedback matrix $\boldsymbol{R}$; feature matrix $\boldsymbol{F}$; parameters $\beta, \lambda_1, \lambda_2, \gamma$
Output: feature weights $\boldsymbol{w}$

$\boldsymbol{w} \leftarrow$ inverted document frequency;
$\boldsymbol{S}^w \leftarrow$ calculate according to Equation (1);
$\boldsymbol{S}_1, \boldsymbol{S}_2 \leftarrow \boldsymbol{S}^w$; $\boldsymbol{V}_1, \boldsymbol{V}_2 \leftarrow \boldsymbol{0}$;
while stopping criterion is met **do**
  // update $\boldsymbol{w}$
  while no convergence of $\boldsymbol{w}$ **do**
    $\nabla\boldsymbol{w} \leftarrow$ calculate according to Equations (2) and (3);
    $\varphi \leftarrow$ heuristically search the step;
    $\boldsymbol{w} \leftarrow P[\boldsymbol{w} - \varphi\nabla\boldsymbol{w}]$;
    $\boldsymbol{S}^w \leftarrow$ calculate according to Equation (1);
  // update $\boldsymbol{S}_1, \boldsymbol{S}_2$
  $\boldsymbol{S}_1 \leftarrow$ calculate according to Equation (4);
  $\boldsymbol{S}_2 \leftarrow$ calculate according to Equation (5);
  // update $\boldsymbol{V}_1, \boldsymbol{V}_2$
  $\boldsymbol{V}_1 \leftarrow \boldsymbol{V}_1 - \beta(\boldsymbol{S}_1 - \boldsymbol{S}^w)$;
  $\boldsymbol{V}_2 \leftarrow \boldsymbol{V}_2 - \beta(\boldsymbol{S}_2 - \boldsymbol{S}^w)$;
  $\beta \leftarrow \beta \cdot \gamma$;

6. EXPERIMENTAL EVALUATION

In this section, we report a series of experiments to evaluate the performance of our proposed method on two different use cases - conference paper recommendation and email recipient recommendation. We first analyze the parameter settings, where the best values are selected through validation, and then compare our method with other 7 top-N recommendation algorithms. Finally, we evaluate the effectiveness of our proposed method on feature selection. We implemented the two proposed algorithms HSLIMtf and HSLIMlf on CUDA[2] to ensure high efficiency of training.

Table 1: The Statistics of Datasets

Dataset	#users	#items	#feeds	density	#features
NIPS	2037	1740	3990	0.11%	13649
Enron1	663	1773	1588	0.14%	25133
Enron2	953	5366	3401	0.07%	32063

6.1 Datasets

We first respectively introduce the datasets as follows:

- **NIPS**[3] contains paper-author and paper-word matrices extracted from co-author network at the NIPS conference over 13 volumes. We regard authors as users, papers as items and the contents of papers as side information. The data has 2037 users (authors) and 1740 items (papers), where 13649 words have been extracted from the corpus of item profiles. The content of the papers is preprocessed such that all words are converted to lower case and stemmed and stop-words are removed. One may note that NIPS dataset is very sparse, e.g., some author may publish only one or two papers, which shows the importance of properly leveraging side information for recommendation.

- **Enron**[4] represents the 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. We consider the two largest mailboxes (dasovich-j and kean-s) and within each mailbox the emails sent by the owner, respectively denoted by Enron1 and Enron2. The messages have been preprocessed by removing the headers (from/to/cc fields), converting all tokens to lower case and removing numbers, stop-words and infrequent tokens (appearing < 5 times).

Thus, we prepared *three* datasets for the experiments, summarized in Table 1, where #users, #items, #feeds and #features, respectively, represent the numbers of users, items, feedbacks and features. The density denotes the sparseness of feedback, defined as:

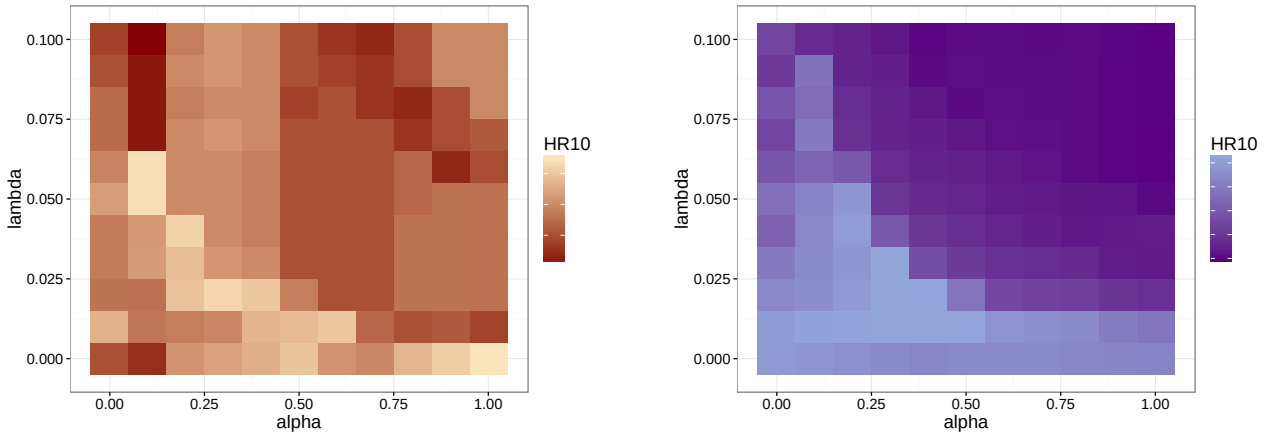
$$density = \frac{\#feeds}{\#users \times \#items}.$$

The main reason for extracting two datasets from Enron is to allow performance comparison under various sparsity and dimensionality, but with the identical assumption of application distribution. We can see from Table 1 that all datasets experimented have fairly sparse feedback. Besides, as shown in the #features column, the three datasets all have very high dimensionality. Particularly, NIPS has more users but less items than Enron1 and Enron2. Meanwhile, in comparison with Enron1, Enron2 is even more sparse and has much higher dimensionality of side information.

6.2 Evaluation Methodology

To comprehensively understand the effectiveness of the methods, we adopt 5-time Leave-One-Out Cross Validation (LOOCV). Specifically, the dataset is split into training and

[2]http://www.nvidia.cn/object/cuda-cn.html
[3]http://www.cs.nyu.edu/~roweis/data.html
[4]https://www.cs.cmu.edu/~enron/



(a) NIPS (b) Enron1

Figure 1: Results of Parameter Selection

test sets by randomly selecting one of the non-zero entries for each user to be part of the test set. The training set is used to train a model, then a size-N ranked list of recommended items for each user is generated. The evaluation of the model is conducted by comparing the recommendation list of each user and the item of that user in the test set.

The recommendation quality is measured by the Hit Rate (HR) and the Average Reciprocal Hit Rank (ARHR) [11]. As pointed out in [22], they are the most direct and meaningful measures in top-N recommendation scenarios. Specifically, HR is defined as:

$$HR = \frac{\#hits}{\#users},$$

where $\#hits$ is the number of users whose item in the test set is recommended, i.e., hit, in the size-N recommendation list, and $\#users$ is the total number of users. A drawback of HR is that it treats all hits equally regardless of where they appear in the top-N list. ARHR addresses it by rewarding each hit based on where it occurs in the top-N list, which is defined as follows:

$$ARHR = \frac{1}{\#users}\sum_{i=1}^{\#hits}\frac{1}{p_i},$$

where p_i is the position of the test item in the ranked top-N list for the i^{th} hit. That is, hits that occur earlier in the ranked list are weighted higher than those occur later, and thus ARHR measures how strongly an item is recommended. In the set of experiments, we set N as 5,10,15 and 20, respectively.

6.3 Parameter Analysis

We first select the value of parameters through validation. As suggested, we start with HSLIMlf for the parameter analysis, and thus there are three parameters to select, respectively, $\alpha, \lambda_1 and \lambda_2$. Naturally, the selection of λ_1 and that of α and λ_2 could be performed separately. In particular, we can first set $\alpha = \lambda_2 = 0$ and select λ_1, and then based on the selection of λ_1, we simultaneously select α and λ_2. We conduct the parameter analysis on the three datasets, and present the selection result on NIPS and Enron1. The result on Enron2 is omitted as it shows the similar result as Enron1.

Table 2: Result of Parameter Analysis

Dataset	Method	λ_1	α	λ_2
NIPS	HSLIMtf	0.01		0.01
Enron1	HSLIMlf	0.01	0.03	0.02
Enron2	HSLIMlf	0.01	0.02	0.01

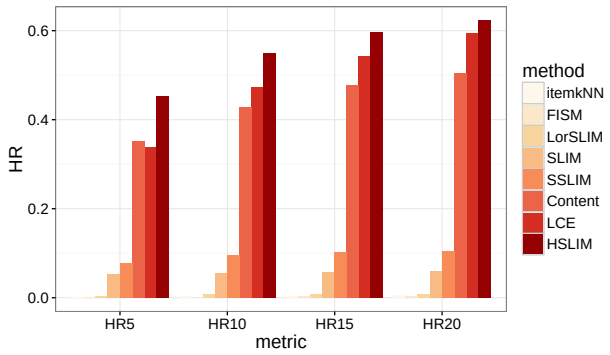
We select λ_1 by varying it from 0.00 to 0.10 and find that it achieves the best result on both datasets when $\lambda_1 = 0.01$. We proceed to evaluate the selection of α and λ_2, the result of which is showing in Figure 1.

Figure 1 shows the result of parameter selection on NIPS and Enron1, where the grids are colored from dark to bright, representing the value of HR10 from low to high. It has been revealed by Figure 1(a) that the result improves with the growth of α and it achieves the best when $\alpha = 1$. The result suggests that the side information of NIPS is qualified and HSLIMtf should be employed to achieve a better result. On the other hand, as shown in Figure 1(b), the best result appears when setting $\alpha = 0.3$ and $\lambda_2 = 0.02$, which suggests the side information of Enron1 is not well correlated with feedback information. The final result of parameter selection is summarized in Table 2.

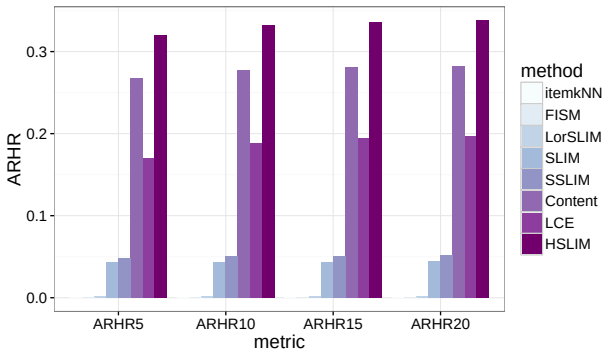
6.4 Performance Comparison

We evaluate the performance of our proposed method on top-N recommendation, where the parameters are set according to Table 2. In this set of experiments, we simply represent our method as HSLIM, where it is HSLIMtf in NIPS and HSLIMlf in Enron1 and Enron2. The proposed method is compared with other seven approaches, including the item neighborhood-based collaborative filtering method itemkNN [11], the state-of-the-art top-N recommendation methods SLIM [22], FISM [18], LorSLIM [8], the hybrid methods LCE [25], SSLIM [23] and the pure content-based recommender Content. We use the Librec library [5](a java library for recommendation) to run itemkNN, SLIM, FISM and we also implement SSLIM using Librec. The pure content-

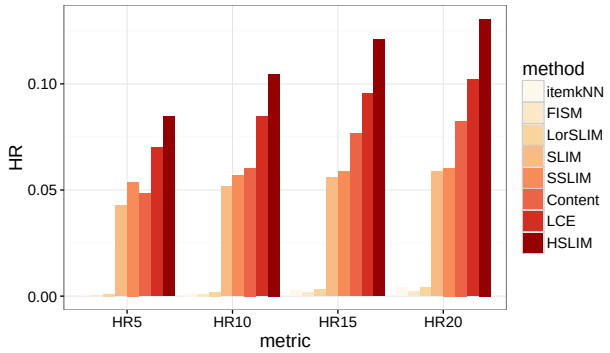
[5]http://www.librec.net



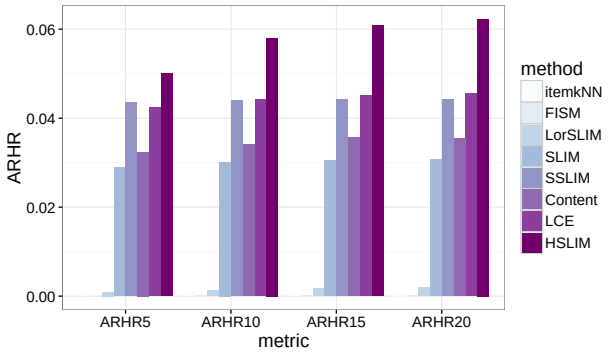
(a) NIPS



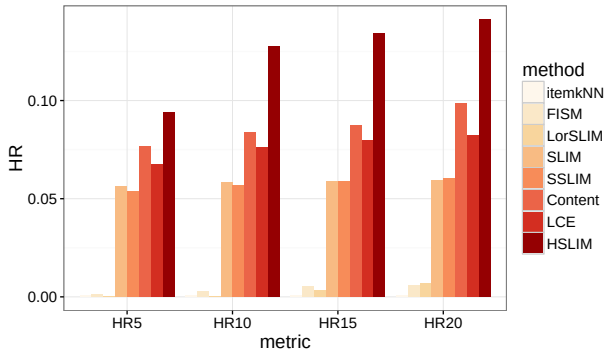
(b) NIPS



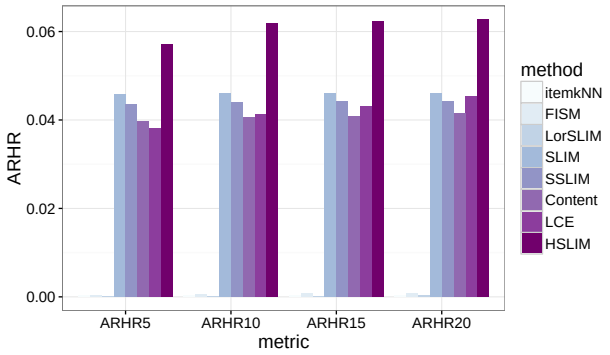
(c) Enron1



(d) Enron1

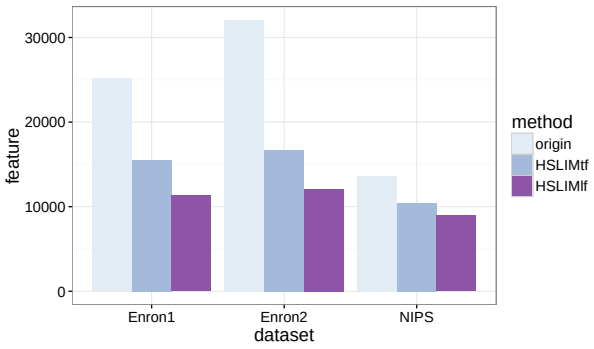


(e) Enron2

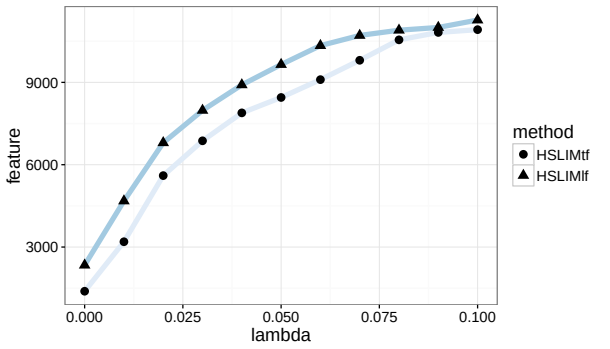


(f) Enron2

Figure 2: Results of Performance Comparison



(a) Enron1, Enron2 and NIPS



(b) NIPS

Figure 3: Results of Feature Selection

based method Content is implemented by setting the feature weights as the *inverted document frequency*, which is the global term weights in text similarity measuring. We depict the results in Figure 2.

Figures 2(a) and 2(b) represent the recommendation performance on NIPS. While the dataset is very sparse, the traditional top-N recommendation methods, including itemkNN, SLIM, FISM, LorSLIM, are showing poor performance. As Content, SSLIM, LCE utilize the side information that is displayed to have good quality for recommendation, the performances of them are much superior. However, affected by the high dimensionality of side information, LCE and SSLIM actually degrade the performance in comparison with Content. Our proposed method HSLIM reduces the redundancy or trivial features, which further improves the performance over LCE on HR and Content on ARHR.

The similar results are revealed by Figures 2(c)–2(f). The difference is, the side information of Enron1 and Enron2 have higher dimensionality, thus the priority of feature selection is more evidently displayed. Specifically, as shown by Figures 2(c) and 2(d), the difference of behavior between the conventional top-N recommender systems (SLIM, FISM, LorSLIM) and the method utilizing side information (Content, SSLIM, LCE) are shortened. We contend the reason is that Content, SSLIM, LCE are heavily affected by the high dimensionality of side information of Enron1, as the number of dimensions is nearly 2 times that of NIPS. Besides, the superiority of HSLIM over the compared methods is more evidently exhibited, especially on the HR metric. Moreover, as displayed by Figures 2(e) and 2(f) on Enron2, while generally the result is similar with that on Enron1, with the increase of sparsity of feedback and dimensionality of side information, the performance of the compared hybrid methods is further affected, especially LCE, and oppositely the superiority of HSLIM is further confirmed.

6.5 Evaluate Feature Selection

Finally, we evaluate the effect of feature selection of HSLIM. In this set of experiments, we first compare the number of features reduced on different datasets, given the selected parameters, to demonstrate the effectiveness of feature selection. The result is depicted in Figure 3(a). We can see from the figure that our proposed method largely reduces the number of features to achieve greater performance. In comparison with HSLIMtf, HSLIMlf can reduce more features as the feature weights $\boldsymbol{w}$ in HSLIMtf is regulated more. Furthermore, the feature selection task is more obviously achieved on Enron2, where almost half of the features are dropped.

Besides, the proposed algorithm HSLIM does not provide a parameter to directly control the number of features to be selected, instead, it encourages more features to be filtered out by penalizing more on $\boldsymbol{w}$. We then analyze the relation between the penalty on $\boldsymbol{w}$ and the feature selection effect. Figure 3(b) shows the result experimented on NIPS, where the lines represent the number of features dropped. When λ_2 grows from 0.00 to 0.05, the linear growth is approximately seen. The growth rate slows down when λ_2 is no less than 0.05 and the gap of reduced features between HSLIMtf and HSLIMlf is also narrowed. In practise, when λ_2 is set higher than 0.05, the problem gets quickly converged and no further chance for features to be filtered out. This has suggested that we can try to select the specific number of features by proportionally increasing the penalty of sparsity on $\boldsymbol{w}$, but the penalization parameter should not be set above a certain threshold (0.05 for λ_2 in this case).

7. CONCLUSION

In this paper, we have presented an embedded feature selection method to utilize high-dimensional side information for top-N recommendation. We propose to learn the weights for features, and introduce sparsity and non-negativity to it. While the feature selection task is achieved by filtering out zero features, the low-rank property is gained as side product through sparsity penalization. In the light of the quality of side information for the recommendation performance, we devise two algorithms, namely HSLIMtf and HSLIMlf, differed in how the feature selection is coupled with learning. To efficiently solve the optimization problems, the Augmented Lagrange Multiplier and Alternating Direction Method are employed. We conduct extensive experiments on different datasets, and the result has demonstrated the superiority of our proposed methods.

8. REFERENCES

[1] D. Agarwal and B. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, Paris, France, June 28 - July 1, 2009*, pages 19–28, 2009.

[2] I. Barjasteh, R. Forsati, F. Masrour, A. Esfahanian, and H. Radha. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pages 91–98, 2015.

[3] I. Barjasteh, R. Forsati, D. Ross, A. Esfahanian, and H. Radha. Cold-start recommendation with provable guarantees: A decoupled approach. *IEEE Trans. Knowl. Data Eng.*, 28(6):1462–1474, 2016.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[5] D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[6] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, Washington, DC, USA, July 25-28, 2010*, pages 333–342, 2010.

[7] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[8] Y. Cheng, L. Yin, and Y. Yu. Lorslim: Low rank sparse linear methods for top-n recommendations. In *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, pages 90–99, 2014.

[9] E. Christakopoulou and G. Karypis. HOSLIM: higher-order sparse linear method for top-n recommender systems. In *Advances in Knowledge*

Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part II, pages 38–49, 2014.

[10] E. Christakopoulou and G. Karypis. Local item-item models for top-n recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016, Boston, MA, USA, September 15-19, 2016*, pages 67–74, 2016.

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

[12] 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.

[13] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, ICDM 2010, Sydney, Australia, 14-17 December 2010*, pages 176–185, 2010.

[14] Y. Gu, B. Zhao, D. Hardtke, and Y. Sun. Learning global term weights for content-based recommender systems. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 391–400, 2016.

[15] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[16] R. He and J. McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 144–150, 2016.

[17] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 507–514, 2005.

[18] S. Kabbur, X. Ning, and G. Karypis. FISM: factored item similarity models for top-n recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 659–667, 2013.

[19] Z. Kang, C. Peng, and Q. Cheng. Top-n recommender system via matrix completion. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 179–185, 2016.

[20] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[21] F. Nie, W. Zhu, and X. Li. Unsupervised feature selection with structured graph optimization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1302–1308, 2016.

[22] 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, December 11-14, 2011*, pages 497–506, 2011.

[23] 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.

[24] S. Roy and S. C. Guntuku. Latent factor representations for cold-start video recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016, Boston, MA, USA, September 15-19, 2016*, pages 99–106, 2016.

[25] 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 - October 06 - 10, 2014*, pages 89–96, 2014.

[26] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD2008, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 650–658, 2008.

[27] S. Tabakhi, P. Moradi, and F. Akhlaghian. An unsupervised feature selection algorithm based on ant colony optimization. *Eng. Appl. of AI*, 32:112–123, 2014.

[28] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, San Diego, CA, USA, August 21-24, 2011*, pages 448–456, 2011.

[29] H. Wang, N. Wang, and D. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015, Sydney, NSW, Australia, August 10-13, 2015*, pages 1235–1244, 2015.

[30] S. Wang, J. Tang, and H. Liu. Embedded unsupervised feature selection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, January 25-30, 2015, Austin, Texas, USA.*, pages 470–476, 2015.

[31] X. Xin, D. Wang, Y. Ding, and C. Lini. FHSM: factored hybrid similarity methods for top-n recommender systems. In *Web Technologies and Applications - 18th Asia-Pacific Web Conference, APWeb 2016, Suzhou, China, September 23-25, 2016. Proceedings, Part II*, pages 98–110, 2016.

[32] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference, ICML 2007, Corvallis, Oregon, USA, June 20-24, 2007*, pages 1151–1157, 2007.

[33] Y. Zheng, B. Mobasher, and R. D. Burke. CSLIM: contextual SLIM recommendation algorithms. In *Eighth ACM Conference on Recommender Systems, RecSys 2014, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 301–304, 2014.