Learning to Recommend Accurate and Diverse Items

Peizhe Cheng[*]
Shandong University
chengpz_sdu@163.com

Shuaiqiang Wang[*]
The University of Manchester
shuaiqiang.wang@manchester.ac.uk

Jun Ma
Shandong University
majun@sdu.edu.cn

Jiankai Sun
The Ohio State University
sun.1306@osu.edu

Hui Xiong
Rutgers University
hxiong@rutgers.edu

ABSTRACT

In this study, we investigate diversified recommendation problem by supervised learning, seeking significant improvement in diversity while maintaining accuracy. In particular, we regard each user as a training instance, and heuristically choose a subset of accurate and diverse items as ground-truth for each user. We then represent each user or item as a vector resulted from the factorization of the user-item rating matrix. In our paper, we try to discover a factorization for matching the following supervised learning task. In doing this, we define two coupled optimization problems, parameterized matrix factorization and structural learning, to formulate our task. And we propose a diversified collaborative filtering algorithm (DCF) to solve the coupled problems. We also introduce a new pairwise accuracy metric and a normalized topic coverage diversity metric to measure the performance of accuracy and diversity respectively. Extensive experiments on benchmark datasets show the performance gains of DCF in comparison with the state-of-the-art algorithms.

Keywords

Diversity; Collaborative filtering; Recommender systems; Structural SVM

1. INTRODUCTION

In recent years, recommender systems have become a de facto standard and a must-own tool for e-commerce to promote business and help customers find products. As an effective technique addressing the information overload problem, recommender systems generate item recommendations from a large collection in favor of user preferences.

With accuracy being the primary concern in recommendation tasks, diversity has been increasingly recognized as a crucial issue [26, 32]. The recommender systems with poor diversity might narrow down the users' horizons and make them frustrated. Actually, broadening users' horizons has become one of the important qualities for recommender systems [20]. A system with broad horizons may provide a win-win situation: users can find more interesting items and e-commerce enterprises can increase their sales and improve users' satisfaction [22].

Recently, the diversity issue in recommender systems has received increasing attention [2, 4, 35]. Generally, these approaches use the heuristic strategy to re-rank the items for recommendation based on certain diversity metric, which mainly involves two steps: (1) generating a candidate set of favorable items based on the accuracy metric, and then (2) selecting k items from the candidates by maximizing the recommendation diversity metric. However, these algorithms either use a limited feature space or require extensive tuning for different parameter settings [33, 34].

In this study, we investigate diversified recommendation problem by supervised learning, targeting significant improvement in recommendation diversity meanwhile maintaining accuracy. In particular, we regard each user associated with a set of rated items as a training instance. The users and items are represented based on the results of the matrix factorization with a given user-item rating matrix.

Theoretically, a matrix factorization process is a projection from a high-dimension space into a smaller one. Given a user-item rating matrix, there could be numerous possible results of the projections, each leading to a representation of the users and items. *In our work, we attempt to discover a factorization for matching the structural learning task.*

In doing this, we formulate two coupled optimization problems to guarantee that the factorization results can match the learning task: (1) With a set of parameters generated by structural support vector machine (SVM), we present parameterized matrix factorization, which can generate the representations of the users and items for structural SVM; (2) With the representations of the users and items generated by parameterized matrix factorization, we utilize structural SVM to output the recommendation model as well as the parameters for parameterized matrix factorization.

We propose a diversified collaborative filtering algorithm (DCF) to solve the coupled problems. In DCF, structural support vector machine (SVM) learns a recommendation model that can predict a set of items for each user, and a parameterized matrix factorization process is integrated in each iteration of the structural SVM for generating representations of users and items. We then adopt the cutting plane method to achieve the optimization results for recommendations. Extensive experiments on benchmark datasets

[*] Both of the authors contributed equally to the paper.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052585

demonstrate the significant performance gains of DCF in comparison with the state-of-the-art algorithms.

Contribution. Our main contributions are summarized as follows.

(1) We investigate learning-based diversified recommendation problem, targeting significant improvement in recommendation diversity meanwhile maintaining accuracy. Then we formulate two coupled optimization problems to discover a factorization for matching the learning task.

(2) We propose DCF, a diversified collaborative filtering algorithm to solve the coupled problems, where a parameterized matrix factorization process is integrated in each iteration of structural SVM for learning.

(3) We introduce a new pairwise accuracy metric and a normalized topic coverage diversity metric to measure the performance of accuracy and diversity respectively, and show the consistency with traditional measures.

Organization. The rest of the paper is organized as follows. Section 2 provides a brief survey of recent related work. Section 3 formulates the diversified recommendation problem. Section 4 proposes the DCF algorithm for simultaneous improvement in accuracy and diversity. Section 5 reports the experimental results. Section 6 concludes the paper.

2. RELATED WORK

2.1 Collaborative Filtering

Existing collaborative filtering (CF) algorithms mainly fall into two categories: *memory-based CF* and the *model-based CF* [3]. In memory-based rating-oriented CF, user-based CF [13, 30] and item-based CF [19, 24] are two representatives, which make predictions by utilizing historical ratings associated with similar users or similar items.

In contrast to memory-based methods, model-based methods try to explain the ratings by characterizing both items and users with vectors of factors inferred from item rating pattern. Some of the most successful realization of latent factor models are based on matrix factorization [18]. Weimer et al. [31] propose CofiRank, which uses maximum margin matrix factorization to optimize ranking of items for collaborative filtering. Shi et al. [25] provide ListRank-MF, which generates features with matrix factorization, and then utilize a listwise learning to rank method to rank the items for recommendations.

2.2 Recommendation Diversity

The recommendation diversity can be viewed at either an aggregate level or an individual level.

Aggregate Diversity. The aggregate diversity of recommendations across all users can be evaluated by absolute long-tail metrics, relative long-tail metrics, and slope of the log-linear relationship between item popularity rank and recommendations [6, 21]. Adomavicius et al. [1, 2] utilize the total number of distinct items among the top-N items recommended across all users as the absolute long-tail metric to measure aggregate diversity. They firstly use a standard ranking approach to maximize the accuracy of recommendations. After the rating estimation is performed, they formulate the problem of diversity maximization as a well-known max-flow or maximum bipartite matching problem in graph theory for optimization.

Individual Diversity. The individual diversity is oriented to each individual user, which attempts to achieve a diverse recommendation result for each target user. A popular individual diversity measure is *topic coverage* [4, 28], which measures the proportion of unique subtopics covered by a given rank or set. For example, Ashkan et al. [4] assume that topic coverage is equivalent to genre coverage for movies and maximize a modular function subject to a submodular constraint. In addition, Parambath et al. [23] define the genre coverage as the average ratio of relevant genres recommended to each user, which is used as the performance metrics for diversity.

2.3 Structural SVM

Structural support vector machine (SVM) generalizes the SVM classifier, which allows training a classifier for general structured output labels such as trees, sequences, or sets [27].

For example, Cui et al. [9, 10] utilize the structure SVM to predict the annotations of the images, where the annotations of the target images are regarded as a set structure, differing from traditional methods that treat each annotation independently. Yue et al. [33] formulate the task of diversified retrieval as the problem of predicting diverse subsets, and adopt Structual SVM to train a diversified model for search engines.

Generally speaking, most conventional structural SVM-based algorithms use n cutting-plane models for optimization, which results in expensive or intractable computational costs in learning a function with complex outputs. To overcome this challenge, Joachims et al. [15] propose the "1-slack" formulations of the structural SVM, which use a single cutting plane model for the sum of the hinge-losses, demonstrating fast training for structural SVM.

3. PROBLEM FORMULATION

Existing diversified algorithms generally utilize diversity metrics for heuristically reranking of the items to make predictions. However, we address a learning-based diversified recommendation problem by treating the recommended items as a set structure, and utilize machine learning techniques to train a recommendation model for predictions.

Matrix factorization (MF) is a very common technique to represent users and items in recommendation tasks. Let U and I be the sets of users and items respectively, where the cardinalities $|U| = m$ and $|I| = n$. $\mathbf{R}_{m\times n}$ is the user-item rating matrix, where each element $r_{u,i} \in \mathbb{N}$ indicates that the item i has been rated by the user u and the rating score is $r_{u,i}$, otherwise indicating that item i has not been rated by the user u yet. Resulting from the MF technique, two low-rank matrices $\mathbf{U}_{k\times m}$ and $\mathbf{I}_{k\times n}$ can be generated to approximate the rating matrix R, i.e., $\mathbf{R}_{m\times n} \approx \mathbf{U}_{k\times m}^{\top}\mathbf{I}_{k\times n}$. Thus both users and items can be represented as k–dimensional vectors, which are the columns of $\mathbf{U}$ and $\mathbf{I}$ respectively.

Theoretically, a matrix factorization process is a projection from a n-dimension space into a k-dimension one. Given a user-item rating matrix, there could be numerous possible results of the projections, each leading to a representation of the users and items. In this paper, we introduce a vector of parameters $\boldsymbol{\sigma}$ into matrix factorization to represent these multiple possible factorization results, which is referred to as *parameterized matrix factorization*. Given a user-item matrix $\mathbf{R}$ and a set of parameters $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, parameterized matrix factorization generates two low-rank matrices $\mathbf{U}_{k\times m}$ and $\mathbf{I}_{k\times n}$ to represent the users and items.

Definition 1. (Parameterized Matrix Factorization). *Given a user-item rating matrix $\mathbf{R}$ and a set of parameters $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, parameterized matrix factorization generates two low-rank matrices $\mathbf{U}_{k\times m}$ and $\mathbf{I}_{k\times n}$:*

$$\underset{\mathbf{U},\,\mathbf{I}}{\arg\min} \quad S(\mathbf{U}, \mathbf{I}; \boldsymbol{\Sigma}, \mathbf{R}), \tag{1}$$

so that

$$\mathbf{R}_{m\times n} \approx \mathbf{U}_{k\times m}^{\top} \boldsymbol{\Sigma}_{k\times k} \mathbf{I}_{k\times n},$$

where $\boldsymbol{\Sigma}$ is a k–dimensional diagonal matrix, and the diagonal elements are $\sigma_1, \sigma_2, \ldots, \sigma_k$.

With the representations of the users and items, we formulate diversified recommendation as a learning task and regard each user as an instance. The diversified recommendation problem tries to discover a set of accurate and diverse items to recommend each user. Let $\mathcal{Y}$ be the set of labels for the training instances. In particular, each $y_u \in \mathcal{Y}$ for user u is a set of vectors $y_u = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_l\}$, where $\mathbf{y}_i \in \mathbf{I}$ is the vector of the ith item that should be recommended to u. Given a user matrix $\mathbf{U}$ and the set of labels $\mathcal{Y}$, learning-based diversified recommendation attempts to discover a factorization for matching the structural learning task.

Definition 2. (Learning-based Diversified Recommendation). *Given a user matrix $\mathbf{U}$ where each column $\mathbf{u} \in \mathbf{U}$ is the vector of user u, an item matrix $\mathbf{I}$ where each column $\mathbf{i} \in \mathbf{I}$ is the vector of item i, a set of labels $\mathcal{Y}$ for users, and a set of possible recommendation functions $\mathcal{H}$, where each $h : \mathbf{U} \times \mathbf{I}^n \to \mathbf{I}^l$ recommending l items to each user based on users' historically rated items, learning-based diversified recommendation attempts to discover a recommendation function h as well as a set of parameters $\boldsymbol{\sigma}$ for optimizing parameterized matrix factorization:*

$$\underset{h\in\mathcal{H},\ \boldsymbol{\sigma}\in\mathbb{R}^k}{\arg\min} \quad L\Big(h(\mathbf{U}, \mathbf{I}^n), \boldsymbol{\Sigma}; \mathcal{Y}, \mathbf{U}, \mathbf{I}\Big) \tag{2}$$

where $L\big(h(\mathbf{U}, \mathbf{I}^n), \boldsymbol{\Sigma}; \mathcal{Y}, \mathbf{U}, \mathbf{I}\big)$ is the loss function with full consideration of the recommendation accuracy and diversity issues for each user u.

As shown in Definitions 1 and 2, we have two coupled optimization problems:

- With a set of parameters $\boldsymbol{\sigma}$ generated by Equation (2) in the learning task, parameterized matrix factorization generates the representations of the users $\mathbf{U}$ and items $\mathbf{I}$ for learning.

- With the representations of the users $\mathbf{U}$ and items $\mathbf{I}$ generated by Equation (1) in parameterized matrix factorization, the learning algorithm outputs the recommendation model as well as the parameters $\boldsymbol{\sigma}$ for parameterized matrix factorization.

4. THE DCF ALGORITHM

In this study, we propose DCF, a learning-based diversified collaborative filtering (CF) algorithm, targeting significant improvement in recommendation diversity while maintaining accuracy.

4.1 Ground-Truth For Training

Since our algorithm performs a supervised learning process, we should generate a label set $\mathcal{Y}$, where each user u is labeled by a set of accurate and diverse items represented as a set of vectors $y_u \in \mathcal{Y}$ as the ground-truth. However, it is too heavy to label the tremendous training instances manually. In this section, we present an automatic heuristic labeling method, which tries to maximize the trade-off between accuracy and diversity.

Our labeling method involves two steps: (I) Filter a set of high-rated items C for each user as a set of candidates, and (II) select a set of items from C by maximizing the diversity measure as the ground-truth of the target user. In Step (I), an item i is high-rated for user u if the rating score $r_{u,i} \geq \gamma \times \overline{r}_u$ holds, where $\overline{r}_u$ is the average rating of u, and $\gamma \geq 0$ controls the balance between accuracy and diversity. In particular, a large γ leads to a very small set of candidates, resulting in high accuracy but low diversity, whereas a small γ leads to a very large set of candidates, resulting in high diversity but low accuracy. For each target user $u \in U$ with a set of candidate items $C = \{i_1, i_2, \ldots, i_c\}$, Step (II) greedily chooses l ($l \leq c$) items from C as the ground-truth y_u by maximizing the F-measure [5] between accuracy $f(y_u)$ and diversity $g(y_u)$:

$$\underset{y_u\in C^k}{\arg\max} \frac{2 \times f(y_u) \times g(y_u)}{f(y_u) + g(y_u)}.$$

We use measures that are defined and described in experimental section to obtain ground truth because the label process should be consistent with the evaluation process. It is reasonable to validate our model with consistent ground truth. In doing this, we can get a training set where each user is labeled by l items. Obviously, the complexity of the selection process is $O(nc^k)$. The complexity is reduced to $O(nck)$, and the approximation ratio is $(1 - 1/e)$ [12]. Inapproximability results show that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for maximum coverage [11].

4.2 Loss Functions

According to Definitions 1 and 2, there are two loss functions: $S(\mathbf{U}, \mathbf{I}; \boldsymbol{\Sigma}, \mathbf{R})$ in the parameterized matrix factorization (MF) task, and $L\big(h(\mathbf{U}, \mathbf{I}^n), \boldsymbol{\Sigma}; \mathcal{Y}, \mathbf{U}, \mathbf{I}\big)$ in the learning task.

Parameterized MF loss. As Definitions 1 mentioned, the Parameterized MF loss function can be defined as follows:

$$\underset{\mathbf{U},\mathbf{I}}{\arg\min} \quad \sum_{\forall r_{u,i}\in\mathbb{R}_0} (r_{u,i} - \mathbf{u}^{\top}\boldsymbol{\Sigma}\mathbf{i})^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{I}\|_F^2), \tag{3}$$

$||\cdot||_F^2$ is the Frobenius 2-norm for avoiding overfitting, and the parameter λ balances the accuracy and the regularization terms.

Note that Equation (3) is very similar to the forms of the singular value decomposition (SVD). However, they are different: (1) The diagonal elements of $\boldsymbol{\Sigma}$ are not the singulars of $\mathbf{R}$. Their values are given as the inputs of the algorithm. (2) $\mathbf{U}$ and $\mathbf{I}$ are not comprised of the left and right singular vectors, either. They are the outputs of the algorithm, which is similar to the optimization of the original MF loss function:

$$\arg\min_{\mathbf{U},\mathbf{I}} \sum_{\forall r_{u,i} \in \mathbb{R}_0} (r_{u,i} - \mathbf{u}^\top \mathbf{i})^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{I}\|_F^2). \quad (4)$$

Learning loss. In the learning process, we treat recommended items as a set structure, and utilize structural support vector machine (SVM) [15] to train a recommendation model. Let $\mathbf{U}$ be a user matrix, where each column $\mathbf{u} \in \mathbf{U}$ represents the vector of user u. Let $\mathbf{I}_u$ be the vectors of the items rated by u, and $y_u \in \mathcal{Y}$ be the ground-truth of the user u. Structured SVM trains a classifier to predict a set of items for each user by optimizing the following loss function:

$$\begin{aligned}
&\arg\min_{\mathbf{w},\boldsymbol{\sigma}} \quad \frac{1}{2}\left(\|\mathbf{w}\|^2 + \|\boldsymbol{\sigma}\|^2\right) + C\xi \\
&\text{s.t.} \quad \forall \mathbf{u} \in \mathbf{U}, \forall y \in \mathcal{Y} \setminus y_u : \qquad (5) \\
&\mathbf{w}^\top \sum_{u \in U} [\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y_u) - \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)] \geq \sum_{u \in U} \Delta(y_u, y) - m\xi
\end{aligned}$$

where $\mathbf{w}$ is the parameters for learning, C is a trade-off parameter, Ψ is the vector of the joint features representing the structure of instances, and Δ is the error function quantifying the distance between a prediction y and the ground-truth y_u for each user u. To apply the algorithm to a specific problem, we have to specify the joint features $\Psi(x, y)$ and the loss function $\Delta(y_u, y)$ in Equation (5).

4.3 Joint Features Ψ

Since the accuracy and diversity metrics have been unified into the same set level, the joint features can be represented straightforwardly with latent features generated by matrix factorization. Given a k–dimensional user vector $\mathbf{u}$, a set of k–dimensional vectors $\mathbf{I}_u$ of the items rated by u, a set of parameters $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, and a prediction y, the vector of the joint features $\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$ are shown as follows:

$$\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y) = \begin{bmatrix} z_1 \sum_{\forall \mathbf{i} \in y,\ \mathbf{j} \in \mathbf{I}_u \setminus y} \left(\mathbf{u}^\top \boldsymbol{\Sigma} \mathbf{i} - \mathbf{u}^\top \boldsymbol{\Sigma} \mathbf{j}\right) \\ z_2 \sum_{\forall \mathbf{i},\mathbf{j} \in y,\ \mathbf{i} \neq \mathbf{j}} \frac{\mathbf{i}^\top \mathbf{j}}{\|\mathbf{i}\| \|\mathbf{j}\|} \\ \|\mathbf{u}\|^2 \\ z_3 \sum_{\forall \mathbf{i} \in y} \|\mathbf{i}\|^2 \\ \mathbf{u} \\ z_4 \sum_{\forall \mathbf{i} \in y} \mathbf{i} \end{bmatrix},$$

where z_1, z_2, z_3 and z_4 are normalization coefficients. $\boldsymbol{\Sigma}$ is a diagonal matrix, where the diagonal elements are $\sigma_1, \sigma_2, \ldots, \sigma_k$ respectively.

There are totally $2k + 4$ dimensions in the formulation of $\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$. The first dimension involves the information of the recommendation accuracy. The second dimension presents the recommendation diversity based on the average similarity between item pairs. The following two dimensions focus on the norms of the target user and the recommended items, and the rest $2k$ dimensions provide the general information of the target user and the recommended items, where k dimensions are represented as the target user vector and the other k dimensions are the summation of item vectors in prediction y.

4.4 Error Function Δ

In Equation (5), the error function $\boldsymbol{\Delta}(y_u, y)$ measures the distance between the ground-truth label y_u and a prediction y, which is based on the definitions of the recommendation accuracy and the diversity metrics.

The accuracy metric. Since the recommender system is to recommend a list of accurate items to each user, each of the recommended items should be more preferred by users than those not recommended [29]. Thus, we choose a pairwise ranking-oriented metric to construct this objective. Let $\mathbf{I}_u$ be the vectors of the items rated by user u, and the cardinality of the set $|\mathbf{I}_u| = n_u$. Given a prediction $y \subseteq \mathbf{I}_u$, the accuracy metric is defined as follows:

$$f(y; \mathbf{u}, \mathbf{I}_u) = \frac{\sum_{\mathbf{i} \in y, \mathbf{j} \in \mathbf{I}_u \setminus y} \left[P(i \succ j; u) - P(j \succ i; u)\right]}{l(n_u - l)}, \quad (6)$$

where l is the number of the recommended items, and $P(i \succ j; u)$ is the preference function indicating that whether u prefers the item i to j or not:

$$P(i \succ j; u) = \begin{cases} 1, & \text{if } r_{u,i} > r_{u,j} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Obviously, the range of $f(y; \mathbf{u}, \mathbf{I}_u)$ is $[-1, 1]$.

The diversity metric. In our study, we utilize the average dissimilarity metric to measure the diversity of a given prediction y for user u, where the items are represented with the results of the parameterized matrix factorization. Let y be a set of item vectors for predictions. The diversity $g(y)$ is defined as the average dissimilarity of all pairs of items in y:

$$g(y) = \frac{\sum_{\mathbf{i},\mathbf{j} \in y, \mathbf{i} \neq \mathbf{j}} d(\mathbf{i}, \mathbf{j})}{\frac{1}{2} l(l - 1)}, \quad (8)$$

where $d(\mathbf{i}, \mathbf{j})$ is the dissimilarity between item vectors $\mathbf{i}$ and $\mathbf{j}$, and the dissimilarity between item vectors $\mathbf{i}$ and $\mathbf{j}$ is defined based on the cosine similarity:

$$d(\mathbf{i}, \mathbf{j}) = -cos(\mathbf{i}, \mathbf{j}) = -\frac{\mathbf{i}^\top \mathbf{j}}{\|\mathbf{i}\| \times \|\mathbf{j}\|}.$$

Easy to know that the range of $g(y)$ is also $[-1, 1]$.

The error function. The error function includes two parts: the accuracy error $err_{acc}(y_u, y)$ and the diversity error $err_{div}(y_u, y)$.

$$\begin{aligned}
err_{acc}(y_u, y) &= f(y_u; \mathbf{u}, \mathbf{I}_u) - f(y; \mathbf{u}, \mathbf{I}_u) \\
err_{div}(y_u, y) &= g(y_u) - g(y).
\end{aligned}$$

Given a user vector $\mathbf{u}$, the vectors of the items $\mathbf{I}_u$ rated by u, and a prediction y, the error function $\boldsymbol{\Delta}(y_u, y)$ can be defined as the trade-off between $err_{acc}(y_u, y)$ and $err_{div}(y_u, y)$ with the formulation of the F-measure:

$$\Delta(y_u, y) = \frac{2 \times err_{acc}(y_u, y) \times err_{div}(y_u, y)}{err_{acc}(y_u, y) + err_{div}(y_u, y)}. \quad (9)$$

4.5 The DCF Algorithm

Given a user-item rating matrix $\mathbf{R}$, we firstly label the ground-truth for each user with the heuristic labeling method, which has been presented in Section 4.1. We then utilize structural SVM to train a recommendation model, where the parameterized matrix factorization process is integrated in each optimization iteration of the structural SVM to discover a factorization for matching the learning task.

In the rest of this section, we will provide the main processes of DCF respectively, including the optimization of the parameterized matrix factorization, the cutting plane method for optimizing structural SVM, and the update of the parameters for parameterized matrix factorization.

Parameterized matrix factorization. We use the gradient descent method to optimize Equation (3). The gradients of the user and item vectors are calculated as follows:

$$\begin{aligned} \forall \mathbf{u} \in \mathbf{U}: \quad \nabla_{\mathbf{u}} S &= \sum_{\forall \mathbf{i} \in \mathbf{I}} \left(\mathbf{u}^{\top} \mathbf{\Sigma} \mathbf{i} - r_{u,i} \right) \mathbf{\Sigma} \mathbf{i} \quad + \lambda \mathbf{u} \\ \forall \mathbf{i} \in \mathbf{I} \quad : \quad \nabla_{\mathbf{i}} S &= \sum_{\forall \mathbf{u} \in \mathbf{U}} \left(\mathbf{u}^{\top} \mathbf{\Sigma} \mathbf{i} - r_{u,i} \right) \mathbf{\Sigma} \mathbf{u} + \lambda \mathbf{i} \end{aligned} \tag{10}$$

Given a learning rate η, the parameterized matrix factorization algorithm can be optimized iteratively based on the gradient descent method.

Note that the loss function of the parameterized matrix factorization (see Equation 3) can be considered as a weighted version of that of the original matrix factorization (see Equation 4). The constant weighting coefficient to each dimension of $\mathbf{u}^{\top}\mathbf{i}$ is generated from the given diagonal matrix $\mathbf{\Sigma}$. Therefore, the existence of convergence of the parameterized matrix factorization is the same as those of the original matrix factorization algorithm.

Algorithm 1: Parameterized matrix factorization

```
Input  : A user-item rating matrix $\mathbf{R}$, a set of
         parameter $\boldsymbol{\sigma}$, the trade-off parameter $\lambda$, and
         the learning rate $\eta$.
Output: The low-rank user and item matrices $\mathbf{U}$ and $\mathbf{I}$.
1 $\mathbf{U}, \mathbf{I} \leftarrow$ Initialize();
2 repeat
3     forall the $r_{u,i} \in \mathbb{R}_0$ do
4         $\mathbf{u} \leftarrow \mathbf{u} - \eta \nabla_{\mathbf{u}} S$
5         $\mathbf{i} \leftarrow \mathbf{i} - \eta \nabla_{\mathbf{i}} S$
6     end
7 until converge;
8 return: $\mathbf{U}, \mathbf{I}$
```

The pseudocodes of the parameterized matrix factorization algorithm are shown in Algorithm 1. After initialization (line 1), the user and item matrices $\mathbf{U}$ and $\mathbf{I}$ are updated iteratively until convergence based on the gradients of the user and item vectors (lines 2–7), which are calculated with Equation (10).

Structural SVM. In structural SVM, since each possible prediction yields a constraint for optimization, one of the most challenging issues for optimization is that the cardinality of the possible predictions $\mathcal{Y}$ for each training instance is exponential with respect to the dimension of instances. Let n be the total number of items, and l be the number of recommended items. For each user, there are $\binom{n}{l}$ constraints in Equation (5). Let m be the number of users. There are totally $m \times \binom{n}{l} = \frac{m \times n!}{l!(n-l)!}$ constraints, which are too many to consider all of them in the optimization process.

Algorithm 2: One optimization iteration of the cutting plane algorithm

```
Input  : A user matrix $\mathbf{U}_{k \times m}$ where each column $\mathbf{u}$ is
         the vector of user $u$; A set of vectors $\mathbf{I}_u$, each
         representing the item rated by $u$; The
         recommendation ground-truth $y_u$ for $u$; The
         set of the working constraints $W$ ; The
         trade-off parameter $C$, and the threshold $\epsilon$.
Output: A vector of weights $\mathbf{w}$ and the working
        constraints $W$
1  foreach column $\mathbf{u}$ of $\mathbf{U}$ do
2      $H(y; \mathbf{w}) \equiv$
       $\Delta(y_u, y) + \mathbf{w}^{\top}\Big[\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y) - \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y_u)\Big]$;
3      $\hat{y}_u \leftarrow \arg\max_{y \in \mathcal{Y}} H(y; \mathbf{w})$; // Find cutting plane
4  end
5  $\xi \leftarrow \max\{0, \frac{1}{m} \sum_{u \in U} \max_{y \in W} H(y; \mathbf{w})\}$;
6  if $\frac{1}{m} \sum_{u \in U} H(\hat{y}_u; \mathbf{w}) > \xi + \epsilon$ then
7      $W \leftarrow W \cup \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_m\}$;
8      $\mathbf{w} \leftarrow$ optimize Eq.(5) with $C$ over $\bigcup W$;
9  end
10 return: $\mathbf{w}$, $W$
```

In our study, we utilize the cutting plane algorithm [15, 16] to optimize the learning problem iteratively. Algorithm 2 shows one iteration of the cutting plane algorithm, which maintains a set of working constraints W to keep track of the selected constraints, each defining a current relaxation. $\mathbf{w}$ is optimized iteratively through all of the users (lines 1–9) for return (line 10). In particular, with each W generated in the previous iteration, the algorithm attempts to discover the cutting plane $\{\hat{y}_u\}_{u \in U}$ (lines 1–4), based on which the most violated constraint for the target user is found (line 5). Then $\{\hat{y}_u\}_{u \in U}$ is added into the working set W if the corresponding constraint exceeds the current value of ξ by more than ϵ. Once a constraint has been added, the solution is re-computed by optimizing Equation (5) with the new set of working constrains W (lines 6–9).

Algorithm 3: Finding cutting plane

```
Input  : A vector of weights $\mathbf{w}$, a user vector $\mathbf{u}$ for
         representing the user $u$, a set of vectors of
         items rated by $u$, the label $y_u$ for $u$
Output: The most violated constraints $\hat{y}_u$ for $u$
$\hat{y}_u \leftarrow \varnothing$;
for $l = 1$ to $k$ do
    $\mathbf{j} \leftarrow \arg\max_{\mathbf{i} \in \mathbf{I}_u, \mathbf{i} \notin \hat{y}} \Delta(y_u, \hat{y}_u \cup \{\mathbf{i}\}) + \mathbf{w}^{\top}\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, \hat{y}_u \cup \{\mathbf{i}\})$;
    $\hat{y}_u \leftarrow \hat{y}_u \cup \{\mathbf{j}\}$;
end
return: $\hat{y}_u$
```

As we discussed above, the number of the constraints is too huge to exhaustively search the cutting plane $\hat{y}_u$ in each iteration of the Algorithm 2, even for a very small value of k. As Algorithm 3 shows, we present a greedy algorithm for approximation, which can iteratively select a constraint y with the highest marginal gain. In the beginning, $\hat{y}_u$ is initialized as empty (line 1), indicating that none of the items is chosen. Then the algorithm iteratively selects the item with highest marginal gain for k times (lines 2–5). This procedure is a special case of the Budgeted Max Coverage problem [17], which is known to have a $(1-1/e)$ approximation bound.

Update of the diagonal matrix $\mathbf{\Sigma}$. With the weights of the classifier $\mathbf{w}$ and the working constraints W generated by the cutting plane algorithm (Algorithm 2), we can reformulate Equation (5) as follows:

$$\arg\min_{\boldsymbol{\sigma}} \quad \frac{1}{2}\left(\|\mathbf{w}\|^2 + \|\boldsymbol{\sigma}\|^2\right) + \frac{C}{m}\max\left\{0, H(\sigma, y)\right\}$$

where

$$\forall y \in W : H(\boldsymbol{\sigma}, y) = \sum_{u \in U} \Delta(y_u, y) + \sum_{u \in U}\left[\mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y) - \mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y_u)\right]$$

Let

$$L(\boldsymbol{\sigma}) = \frac{1}{2}\left(\|\mathbf{w}\|^2 + \|\boldsymbol{\sigma}\|^2\right) + \frac{C}{m}\max\left\{0, H(\sigma, y)\right\}$$

When $\max H(\boldsymbol{\sigma}, y) > 0$, L is a non-differentiable objective function. Our method utilizes the subgradient method in place of the gradient method for convex optimization problems.

$$L(\boldsymbol{\sigma}) = \frac{1}{2}\left(\|\mathbf{w}\|^2 + \|\boldsymbol{\sigma}\|^2\right) + \frac{C}{m}\left\{\sum_{u \in U} \Delta(y_u, y_m) + \sum_{u \in U}\left[\mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y_m) - \mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y_u)\right]\right\}$$

where

$$y_m = \arg\max_{y \in W} H(\sigma, y).$$

Thus L becomes differentiable, and the gradient of L with respect to $\boldsymbol{\sigma}$ can be calculated.

The pseudocodes of the DCF algorithm. As we have already introduced the main processes of the DCF, we summarize the pseudocodes of the DCF in Algorithm 4, where the outer loop is guaranteed to halt within a polynomial number of iterations [33] (See [15] for the detailed proof).

Algorithm 4: The DCF Algorithm

```
   Input : A user-item rating matrix $\mathbf{R}$
   Output: A vector of weights $\mathbf{w}$ for Structural SVM
           classifier, a user matrix $\mathbf{U}_{k\times m}$, a item matrix
           $\mathbf{I}_{k\times n}$ and a diagonal matrix $\mathbf{\Sigma}$
 1 $\mathcal{Y} \leftarrow$ GenerateLabels($\mathbf{R}$);
 2 $\mathbf{w}, \mathbf{\Sigma} \leftarrow$ Initialize();
 3 $\mathbf{U}, \mathbf{I} \leftarrow$ OptimizeParaMF($\mathbf{R}, \mathbf{\Sigma}$);
 4 $W \leftarrow \varnothing$;
 5 repeat
 6     $\mathbf{w}, W, \hat{Y} \leftarrow$ CutPlane($\mathbf{U}, \mathbf{I}, \mathbf{w}, \mathbf{\Sigma}, \mathcal{Y}, W$);
 7     $\mathbf{\Sigma} \leftarrow$ Update($\mathbf{U}, \mathbf{I}, \mathbf{w}, \mathbf{\Sigma}, W, \hat{Y}$);
 8     $\mathbf{U}, \mathbf{I} \leftarrow$ OptimizeParaMF($\mathbf{R}, \mathbf{\Sigma}$);
 9 until converge;
10 return: $\mathbf{w}$, $\mathbf{U}$, $\mathbf{I}$, $\mathbf{\Sigma}$
```

In Algorithm 4, line 1 generates the recommendation groundtruth for each user. After initialization of the vector of the classifier parameters $\mathbf{w}$ and the diagonal matrix $\mathbf{\Sigma}$ (line 2), parameterized matrix factorization generates the low-rank user and item matrices $\mathbf{U}$ and $\mathbf{I}$ for learning (line 3). After initializing the set of the working constraints as $\varnothing$ (line 4), the structural SVM iteratively learns the recommendation model (lines 5–9) for return (line 10), where line 6 updates $\mathbf{w}$ by performing an optimization iteration of the cutting plane method, based on which line 7 updates the diagonal matrix $\mathbf{\Sigma}$ with gradient descent. Then line 8 generates the user and item matrices by parameterized matrix factorization for the next iteration of the learning process to guarantee that the factorization results can match the learning task.

4.6 Generating Recommendations

Given the vector of the joint features $\Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$, the ultimate goal of the structural SVM is to predict a hypothesis function $h(\mathbf{w}, \mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$ to minimize Equation (5), where $\mathbf{w}$ is the vector of the weights in the structural model. Let $\mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$ be the discriminant function, which can be used to evaluate the prediction y, and the hypothesis function $h(\mathbf{w}, \mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y)$ can be used to predict y by maximizing the discriminant function:

$$h(\mathbf{w}, \mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y) = \arg\max_{y} \mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y) \tag{11}$$

We may notice that the generation process of the recommendation results is very similar to the problem of finding the most violated constraint. It is too time-consuming for us to discover the best sets of the recommendations with $\binom{m\times n}{l}$ constraints. In this study, we still utilize the greedy selection algorithm to generate recommendation lists:

$$\mathbf{j} \leftarrow \arg\max_{\mathbf{i} \in \mathbf{I}_u, \mathbf{i} \notin y} \mathbf{w}^\top \Psi(\mathbf{u}, \mathbf{I}_u, \boldsymbol{\sigma}, y \cup \{\mathbf{i}\}) \tag{12}$$

Items in our recommendation list are ranked according to their selected orders in algorithm 3. The recommendation generation process is also a case of the Budgeted Max Coverage Problem [17] with the approximation bound of $(1-1/e)$.

5. EXPERIMENTS

5.1 Datasets

Since we utilize topic coverage as the diversity metric, the chosen benchmark datasets have to satisfy a fundamental requirement: each item should be associated with *topic* information (e.g., genres of movies in movie recommendation tasks) for labelling training instances and evaluating our predictions.

In this study, we use two movie recommendation datasets in our experiments: MovieLens 100K (ML-100K) and MovieLens 1M (ML-1M) [1]. ML-100K consists of 100,000 ratings on 1,700 movies from 1,000 users, and ML-1M contains 1,000,000

[1] http://www.grouplens.org/node/73

ratings on 4,000 movies from 6,000 users. The rating scales are from 1 to 5, where 5 means the most favorite and 1 means the least favorite. Table 1 lists the statistics of the two datasets.

Table 1: MovieLens 100K and 1M Dataset

	ML-100K	ML-1M
Number of users	943	6,040
Number of movies	1,682	3,952
Number of ratings	100,000	1,000,209
Number of all genres	19	18
Average number of genres	1.7	1.6
Rating scales	1–5	1–5

5.2 Baselines

First of all, we choose MF-pop [2] as our main baseline, which is a diversified matrix factorization (MF) algorithm based on the heuristically re-ranking strategy. Besides, since DCF is a MF-based collaborative filtering algorithm, we also choose the basic MF algorithm (MF-basic) [18] and other MF-based algorithms as our comparison partners, including CofiRank [31] and ListRank-MF [25].

5.3 Evaluation Measures

As our algorithm DCF aims to recommend accurate and diverse items to users, we evaluate DCF from the perspectives of accuracy and diversity to demonstrate its promising performance in our experiments. Furthermore, we also utilize the F–measure [5] to assess the trade-off performance with consideration of both accuracy and diversity.

5.3.1 Accuracy

In this study, we use the normalized discounted cumulative gain ($NDCG$) [14], a listwise accuracy metric for evaluation. Besides, we also propose a normalized pairwise accuracy metric PA to measure the performance of our algorithm based on the accuracy function used in our learning process, as defined in Equation (6).

Listwise Accuracy Metric *NDCG*. $NDCG$ is a standard metric for the document ranking problem in information retrieval, where documents are assigned graded values rather than binary relevance judgments. In recommender systems, item ratings assigned by users can naturally serve as graded relevance judgments. The $NDCG$ at the k-th position with respect to the given user u is defined as follows:

$$NDCG_u@k = z_u \sum_{i=1}^{k} \frac{2^{r_{u,i}} - 1}{\log(1+i)}. \tag{13}$$

For the set of users U with m users, the average $NDCG$ at the k-th position is:

$$NDCG_{avg}@k = \frac{1}{m} \sum_{\forall u \in U} NDCG_u@k.$$

Pairwise Accuracy Metric *PA*. DCF is a set-oriented algorithm, and the order is just generated by the greedy selection algorithm. However, $NDCG$ is very sensitive to the ranking, so we introduce a normalized pairwise accuracy metric PA which measures how accurate the recommended items are in comparison with the other items. Let I_u be the vectors of the test items for user u, and the cardinality of the set $|I_u| = n_u$. Given a prediction $y \subseteq I_u$, the pairwise accuracy metric is defined as follows:

$$PA_u(y) = \frac{\sum_{i \in y, j \in I_u \setminus y} P(i \succ j; u)}{l(n_u - l)}, \tag{14}$$

where l is the number of the recommended items, and $P(i \succ j; u)$ is the preference function indicating that whether u prefers the item i to j or not, which is defined in Equation (7).

Let U be the set of users in the test dataset, and the cardinality is $|U| = m$. The average pairwise accuracy for all of the users can be evaluated as follows:

$$PA = \frac{1}{m} \sum_{\forall u \in U} PA_u(y).$$

Obviously, the range of PA is $[0, 1]$.

5.3.2 Diversity

We utilize α-$NDCG$ to evaluate our algorithms, and introduce the normalized topic coverage NTC as a new diversity Metric.

***α*-*NDCG*.** α-$NDCG$ is an effective diversity evaluation measure for rewarding newly found topics meanwhile penalizing redundant ones [7, 8], which is very similar to the definition of $NDCG$ in Equation (13):

$$\alpha\text{-}NDCG_u@k = z_n \sum_{i=1}^{k} \frac{G_u@i}{\log(1+i)}, \tag{15}$$

where $G_u@k$ is the gain at the k-th position with respect to the given user u. In particular, $G_u@k$ is defined as follows:

$$G@k = \sum_{\forall t \in T} (1-\alpha)^{c_{t,i}-1},$$

where $c_{t,i}$ is the number of times that topic t has appeared in the ranking of the recommendation list up to (and including) the ith position.

Normalized Topic Coverage Diversity Metric *NTC*. α-$NDCG$ represents the improvement of diversity when a new item is added into recommendation, but we also need a set-oriented diversity metric NTC to show the topic coverage in the whole recommendation list. As discussed previously, topic coverage cov is defined as the ratio of the topics (categories) covered by the set of the predictions to the number of all possible topics. However, the maximum value of the topic coverage is generally far less than 1, especially when the number of recommendations k is far less than the number of the possible topics. Meanwhile, it is unreasonable for the recommendation list to cover none of the topics, because each item (movie) is associated with at least one topic(genre). Thus, we present a novel normalized topic coverage diversity metric NTC, which uses the *Min-Max Normalization* method to linearly map the values of topic coverages into $[0, 1]$. Formally, the normalized topic coverage for a target user u and a prediction y is defined as follows:

$$NTC_u(y) = \frac{cov_u(y) - mincov_u}{maxcov_u - mincov_u}, \tag{16}$$

where $cov_u(y)$ is the ratio of topics(genres) covered by the recommendation list y for user u. $maxcov_u$ and $mincov_u$ are the possibly maximal and minimal topic coverages for user u in the test set. For the set of users U with m users, the normalized topic coverage for all users is defined as follows:

$$NTC = \frac{1}{m} \sum_{\forall u \in U} NTC_u(y).$$

Obviously, the range of NTC is $[0, 1]$.

5.3.3 F–Measure

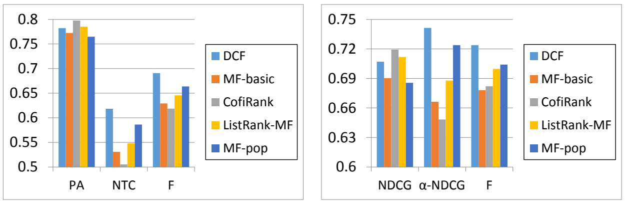
In our study, we present two F-measures for validation: the first one F_{AC} evaluates the trade-off between the mean pairwise accuracy PA and the normalized topic coverage diversity metric NTC, and the second one F_{NDCG} evaluates the trade-off between $NDCG$ and α-$NDCG$.

F-measure [5] for document retrieval problem is defined as the harmonic mean of precision and recall. Similarly, the accuracy and diversity have a similar correlation in our problem. Since the ranges of PA and NTC are both [0,1], and $NDCG$ and α-$NDCG$ are also in the same range, the F-measure can also be used to evaluate the trade-off performance of the recommendation algorithms:

$$F\text{-}measure = \frac{2 * accuracy * diversity}{accuracy + diversity}, \tag{17}$$

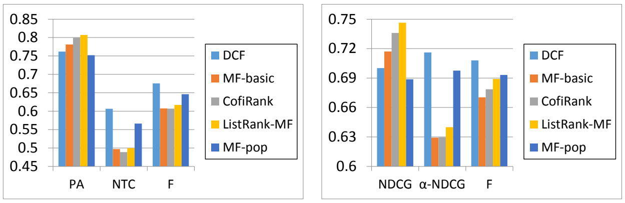
where $accuracy$ can be PA or $NDCG$ and $diversity$ can be NTC or α-$NDCG$.

5.4 Experimental Results



(a) PA and NTC (b) $NDCG$ and α-$NDCG$

Figure 1: Performance on ML-100K



(a) PA and NTC (b) $NDCG$ and α-$NDCG$

Figure 2: Performance on ML-1M

We consider the top-3 recommendation problem, where a list of 3 items is recommended for each user. It is because that each movie has almost 2 genres, but the total number of genres is less than 20 (See Table 1). If the recommendation list contains a large number of items, it is very easy for the recommendation list to cover a large number of topics for most users, which will make the diversity metric lose its significance. Figure 1 and 2 demonstrate the performance of DCF in accuracy, diversity and F-measures on the ML-100K and ML-1M datasets, and the significant difference with respect to DCF is shown in Table 2 and Table 3. Statistical significance of observed differences between two comparisons is tested using a two-tailed paired t-test, and it is denoted using ▲ (or ▼) for strong significant differences for $\alpha = 0.01$; or △ (or ▽) for weak significance for $\alpha = 0.05$. From the table we can see that most of our performance gains on diversity and F-measures are strongly significant. From the figures and the tables we can see that:

(1) The diversified recommendation algorithms can significantly improve recommendation diversity, but DCF and MF-pop cannot avoid loss of accuracy resulting from consideration of the diversity issue. In our experiments, DCF demonstrates significantly greater improvement with less accuracy loss. For example, DCF obtains 22.4% and 12.8% improvement on NTC for the two datasets respectively, with 1.9% and 0.9% loss on PA correspondingly, compared with the best performing traditional algorithm. Similarly, DCF achieves 14.4% and 5.3% improvement in α-$NDCG$ while 1.7% and 0.7% reduction in $NDCG$ with the same comparison as above.

(2) Our learning-based diversified algorithm beats the heuristically re-ranking based algorithm in both accuracy and diversity. For ML-100K, DCF outperforms MF-pop by 5.4% in terms of NTC and 2.4% in terms of α-$NDCG$, and the improvement of our method in PA and $NDCG$ is 2.2% and 3.1% respectively. For ML-1M, our method obtains 7.6% (NTC), 1.9% (α-$NDCG$), 3.1% (PA) and 4.3% ($NDCG$) improvement, compared with MF-pop.

Table 4: Pearson correlation coefficienton between metrics on ML-100K

	PA&$NDCG$	NTC&α-$NDCG$
DCF	0.8275	0.7893
MF-basic	0.8183	0.7906
Cofirank	0.829	0.8042
Listrank-MF	0.8357	0.8073
MF-pop	0.8176	0.8116

Table 5: Pearson correlation coefficienton between metrics on ML-1M

	PA&$NDCG$	NTC&α-$NDCG$
DCF	0.8304	0.795
MF-basic	0.8446	0.792
Cofirank	0.8303	0.7736
Listrank-MF	0.8558	0.796
MF-pop	0.8506	0.7915

(3) DCF makes the best trade-off between accuracy and diversity, compared with MF-basic,CofiRank, ListRank-MF and MF-pop. For example, for ML-100K, DCF obtains 9.8%, 11.6%, 7% and 4.1% improvement respectively on F_{AC}, and 6.7%, 6.1%, 3.5% and 2.8% improvement on F_{NDCG}; For ML-1M, DCF gains 8.8%, 8.9%, 7.1% and 5.8% improve-

$ML-100K$	PA	NTC	F_{AC}	$NDCG$	$\alpha-NDCG$	F_{NDCG}
DCF	0.7823	**0.6184**	**0.6908**	0.707	**0.7414**	**0.7238**
$MF-basic$	0.7723	0.5307$^{\blacktriangledown}$	0.6291$^{\blacktriangledown}$	0.6903	0.6663$^{\blacktriangledown}$	0.6781$^{\blacktriangledown}$
$CofiRank$	**0.7978**	0.5052$^{\blacktriangledown}$	0.6186$^{\blacktriangledown}$	**0.7193**	0.6483$^{\blacktriangledown}$	0.682$^{\blacktriangledown}$
$ListRank-MF$	0.7852	0.548$^{\blacktriangledown}$	0.6455$^{\blacktriangledown}$	0.7117	0.6879$^{\blacktriangledown}$	0.6996$^{\blacktriangledown}$
$MF-pop$	0.7649$^{\triangledown}$	0.5863$^{\blacktriangledown}$	0.6638$^{\blacktriangledown}$	0.6855$^{\triangledown}$	0.7238$^{\triangledown}$	0.7041$^{\blacktriangledown}$

Table 2: Significant Differences with respect to DCF on ML-100K(row with shaded background)

$ML-1M$	PA	NTC	F_{AC}	$NDCG$	$\alpha-NDCG$	F_{NDCG}
DCF	0.7994	**0.5632**	**0.6609**	0.7415	**0.6735**	**0.7059**
$MF-basic$	0.7812$^{\blacktriangledown}$	0.4971$^{\blacktriangledown}$	0.6076$^{\blacktriangledown}$	0.7171$^{\blacktriangledown}$	0.6293$^{\blacktriangledown}$	0.6703$^{\blacktriangledown}$
$CofiRank$	0.7999	0.4887$^{\blacktriangledown}$	0.6067$^{\blacktriangledown}$	0.7359	0.6296$^{\blacktriangledown}$	0.6786$^{\blacktriangledown}$
$ListRank-MF$	**0.8069**$^{\vartriangle}$	0.4994$^{\blacktriangledown}$	0.617$^{\blacktriangledown}$	**0.7464**	0.6399$^{\blacktriangledown}$	0.6891$^{\blacktriangledown}$
$MF-pop$	0.775$^{\blacktriangledown}$	0.5234$^{\blacktriangledown}$	0.6248$^{\blacktriangledown}$	0.7111$^{\blacktriangledown}$	0.6612$^{\blacktriangledown}$	0.6852$^{\blacktriangledown}$

Table 3: Significant Differences with respect to DCF on ML-1M(row with shaded background)

ment on F_{AC}, and 5.3%, 4%, 2.4% and 3% improvement on F_{NDCG}.

(4) The new metrics PA and NTC introduced in Section 5.3 are highly consistent with the traditional measures $NDCG$ and α-$NDCG$ respectively on both datasets. Table 4 and Table 5 show the Pearson correlation coefficienton between accuracy metrics and diversity metrics introduced in Section 5.3 with each method. The more gains a method gets in PA and NTC, the better it performs in $NDCG$ and α-$NDCG$. For example, DCF outperforms other four baselines in both NTC and α-$NDCG$. On the other hand, MF-pop is the worst method in PA and $NDCG$.

6. CONCLUSION AND FUTURE WORK

In this study, we investigate the diversified recommendation problem, targeting significant improvement in recommendation diversity while maintaining accuracy. Different from existing algorithms which generally adopt a heuristic re-ranking strategy with certain diversity metric, we regard diversified recommendation as a supervised learning task, and define two coupled optimization problems to formulate the learning-based diversified recommendation task. We then propose a diversified collaborative filtering algorithm (DCF) to solve the problems, where the items recommended to each user are recognized as a set structure. Comprehensive experiments have validated the effectiveness of our algorithm.

There are several interesting directions for the future work. Firstly, more discriminants, loss functions and joint features can be used for further improvement in recommendation accuracy and diversity. Secondly, we plan to consider other important evaluation issues such as recommendation novelty in DCF to predict high quality results. Last but not least, it is essential to apply DCF to other application domains beyond recommendation for validation purposes.

Acknowledgements. This work is supported by the Natural Science Foundation of China (61672322, 61672324, 71402-083, 61502258), the Natural Science Foundation of Shandong province (2016ZRE27468), the Fundamental Research Funds of Shandong University, the Shandong Province Higher Educational Science and Technology Program (J15LN56), the National Science Foundation of United States under grant CCF-1645599 and IIS-1550302. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

7. REFERENCES

[1] G. Adomavicius and Y. Kwon. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *DiveRS*, pages 3–10, 2011.

[2] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.*, 24(5):896–911, 2012.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[4] A. Ashkan, B. Kveton, S. Berkovsky, and Z. Wen. Optimal greedy diversity for recommendation. In *IJCAI*, pages 1742–1748, 2015.

[5] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, USA, 1999.

[6] E. Brynjolfsson, Y. Hu, and M. D. Smith. Research commentary–long tails vs. superstars: The effect of information technology on product variety and sales concentration patterns. *Info. Sys. Res.*, 21(4):736–747, 2010.

[7] P. Chandar and B. Carterette. Preference based evaluation measures for novelty and diversity. In *SIGIR*, pages 413–422, 2013.

[8] C. L.A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[9] C. Cui, J. Ma, T. Lian, Z. Chen, and S. Wang. Improving image annotation via ranking-oriented neighbor search and learning-based keyword propagation. *J. Assoc. Inf. Sci. Technol.*, 66(1):82–98, 2015.

[10] C. Cui, J. Ma, S. Wang, S. Gao, and T. Lian. Semantically coherent image annotation with a learning-based keyword propagation strategy. In *CIKM*, pages 2423–2426, 2012.

[11] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[12] D. S. Hochbaum, editor. *Approximation algorithms for NP-hard problems.* PWS Publishing, Boston, MA, USA, 1997.

[13] S. Huang, S. Wang, T.-Y. Liu, J. Ma, Z. Chen, and J. Veijalainen. Listwise collaborative filtering. In *SIGIR*, pages 343–352, 2015.

[14] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[15] T. Joachims, T. Finley, and C. Yu. Cutting-plane training of structural svms. *Mach. Learn.*, 77(1):27–59, 2009.

[16] J. E Kelley, Jr. The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.*, 8(4):703–712, 1960.

[17] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

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

[19] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7(1):76–80, 2003.

[20] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI*, pages 1097–1101, 2006.

[21] G. Oestreicher-Singer and A. Sundararajan. Recommendation networks and the long tail of electronic commerce. *MIS Q.*, 36(1):65–84, 2012.

[22] K. Onuma, H. Tong, and C. Faloutsos. TANGENT: a novel, 'surprise me', recommendation algorithm. In *KDD*, pages 657–666, 2009.

[23] S.P. Parambath, N. Usunier, and Y. Grandvalet. A coverage-based approach to recommendation diversity on similarity graph. In *RecSys*, pages 15–22, 2016.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[25] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys*, pages 269–272, 2010.

[26] I. Szpektor, Y. Maarek, and D. Pelleg. When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *WWW*, pages 1249–1260, 2013.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.

[28] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *RecSys*, pages 209–216, 2014.

[29] S. Wang, J. Sun, B. J. Gao, and J. Ma. Adapting vector space model to ranking-based collaborative filtering. In *CIKM*, pages 1487–1491, 2012.

[30] S. Wang, J. Sun, B. J. Gao, and J. Ma. Vsrank: A novel framework for ranking-based collaborative filtering. *ACM Trans. Intell. Syst. Technol.*, 5(3):51:1–51:24, 2014.

[31] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. CofiRank: Maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

[32] L. Wu, Q. Liu, E. Chen, N.J. Yuan, G. Guo, and X. Xie. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM TIST*, 7(3):39, 2016.

[33] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *ICML*, pages 1224–1231, 2008.

[34] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*, pages 123–130, 2008.

[35] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, pages 22–32, 2005.