

Dual-Regularized One-Class Collaborative Filtering

Yuan Yao
Nanjing University, China
yyao@smail.nju.edu.cn

Hanghang Tong
Arizona State University, USA
htong6@asu.edu

Guo Yan, Feng Xu
Nanjing University, China
gyan@smail.nju.edu.cn
xf@nju.edu.cn

Xiang Zhang
CASE Western Reserve
University, USA
xiang.zhang@case.edu

Boleslaw K. Szymanski
Społeczna Akademia Nauk,
Łódź, Poland and RPI, USA
szymab@rpi.edu

Jian Lu
Nanjing University, China
lj@nju.edu.cn

ABSTRACT

Collaborative filtering is a fundamental building block in many recommender systems. While most of the existing collaborative filtering methods focus on *explicit, multi-class* settings (e.g., 1-5 stars in movie recommendation), many real-world applications actually belong to the *one-class* setting where user feedback is *implicitly* expressed (e.g., views in news recommendation and video recommendation). The main challenges in such one-class setting include the *ambiguity* of the unobserved examples and the *sparseness* of existing positive examples.

In this paper, we propose a dual-regularized model for one-class collaborative filtering. In particular, we address the ambiguity challenge by integrating two state-of-the-art one-class collaborative filtering methods to enjoy the best of both worlds. We tackle the sparseness challenge by exploiting the side information from both users and items. Moreover, we propose efficient algorithms to solve the proposed model. Extensive experimental evaluations on two real data sets demonstrate that our method achieves *significant* improvement over the state-of-the-art methods. Overall, the proposed method leads to 7.9% - 21.1% improvement over its best known competitors in terms of prediction accuracy, while enjoying the linear scalability.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database applications—*Data mining*

Keywords

Recommender systems, one-class collaborative filtering, dual regularization

1. INTRODUCTION

Recommender systems have become increasingly indispensable in many applications including movie recommendation [16], hashtag recommendation [2], music recommendation [5], news recom-

mendation [6], etc. Collaborative filtering, which aims at predicting the preferences of users towards items based on the historical user feedback, plays a central role in these recommender systems. To reflect the users' preferences for items, the feedback can be *explicitly* expressed as different ratings (e.g., 1-5 stars in Netflix). The vast majority of existing work focuses on such explicit, *multi-class* recommendation problem. However, in many real situations, the feedback could be *implicitly* expressed with examples like views of news, clicks of webpages, purchases of products, downloads of music, etc. Recommendation with implicit feedback naturally forms the *one-class* collaborative filtering (OCCF) problem [24].

Despite its importance and ubiquity, there is much sparser literature on OCCF, compared with the extensive machinery for multi-class collaborative filtering (MCCF). This is mainly due to the following two challenges. The first challenge rises from the ambiguity of the unobserved data: the unobserved data is not necessarily negative examples, but a mixing of negative examples and missing positive examples. Different from MCCF, which often focuses on the observed data only, special treatments are needed for the missing/unobserved examples in OCCF. Existing solutions for OCCF differ in terms of how they treat the unobserved data, including weighting-based, imputation-based, and sampling-based methods (see Section 6 for a review). While each of them has its own rationality and advantage, it is unclear how to integrate them together to maximally improve the recommendation performance.

Another challenge of OCCF lies in data sparseness (i.e., only an extremely small percentage of data is labeled as positive examples). This challenge compromises the full power of collaborative filtering and leads to the so-called cold-start problem, i.e., it is difficult to make satisfactory recommendations for the cold-start users and/or cold-start items. To tackle this challenge in the MCCF setting, several researchers propose to incorporate the side information such as the demographical information about users, item content, and the social relationships between users (see Section 6 for a review). However, it is unclear how to migrate these techniques to the one-class setting, especially when there is side information from both users and items.

In this paper, we propose a dual-regularized model for one-class collaborative filtering. In particular, we address the ambiguity challenge by integrating two state-of-the-art one-class collaborative filtering methods (weighting-based and imputation-based methods) to enjoy the best of both worlds. The intuition behind this integrated method is two-fold. First, notice that users might prefer a variety of items including those unseen ones. Thus, we impute the unobserved data to indicate the probability that a user would prefer an unseen item. Second, we assign a weight for the imputed data

to indicate its reliability. Furthermore, we tackle the sparseness challenge by encoding the side information from both users and items as two graph regularization terms. Such a treatment (dual-regularization) could help alleviate the cold-start problem and improve the overall performance of recommender systems as well.

In summary, the main contributions of this paper include:

- A unified model for one-class collaborative filtering, which integrates the imputation-based method, the weighting-based method, and the dual regularization from both users and items. Moreover, we propose efficient algorithms to solve the proposed model, and analyze their optimality, convergency, and computational complexity.
- Experimental evaluations on two real data sets to demonstrate the effectiveness and efficiency of the proposed method. For example, our proposed algorithm (*wiZAN-Dual*) outperforms the best known competitors by 7.9% - 21.1% in terms of prediction accuracy. In the meanwhile, *wiZAN-Dual* scales linearly wrt the size of the input data set.

The rest of the paper is organized as follows. Section 2 defines the problem. Section 3 and Section 4 present the proposed formulations and algorithms, respectively. Section 5 presents the experimental results. Section 6 reviews related work, and Section 7 concludes the paper.

2. PROBLEM DEFINITIONS AND PRELIMINARIES

2.1 Problem Definitions

Table 1 lists the main symbols we use throughout the paper. Following conventions, we use bold capital letters for matrices. For example, we use an $m \times n$ matrix \mathbf{R} to denote the user-item feedback, where $\mathbf{R}(u, i) = 1$ if user u had provided positive feedback on item i , and $\mathbf{R}(u, i) = 0$ if the feedback is unknown/unobserved. Without loss of generality, we use u and v to index users, and we use i and j to index items. We use calligraphic font \mathcal{O} and \mathcal{U} to

Table 1: Symbols.

Symbol	Definition and Description
\mathbf{R}	the implicit, one-class feedback matrix
\mathbf{W}	the weighting matrix for \mathbf{R}
\mathbf{P}	the imputation matrix for \mathbf{R}
\mathbf{F}, \mathbf{G}	the low-rank approximations for \mathbf{R}
\mathbf{M}, \mathbf{N}	the user-user graph and the item-item graph
\mathbf{R}'	the transpose of matrix \mathbf{R}
$\mathbf{R}(u, :)$	the u^{th} row of \mathbf{R}
$\mathbf{R}(u, i)$	the element at the u^{th} row and i^{th} column of \mathbf{R}
$\mathbf{I}^{\mathcal{O}}, \mathbf{I}^{\mathcal{U}}$	the indicator matrices for observed and missing data
$\mathbf{I}^{\mathcal{A}}$	the full indicator matrix $\mathbf{I}^{\mathcal{A}} = \mathbf{I}^{\mathcal{O}} + \mathbf{I}^{\mathcal{U}}$
\mathcal{O}	the set of observed data between users and items
\mathcal{U}	the set of unobserved data between users and items
m, n	the number of users and items
u, v	the users
i, j	the items
w	the global weight for \mathbf{W}
p	the global imputation value for \mathbf{P}
r	the low rank for \mathbf{F} and \mathbf{G}
l	the maximum iteration number
ξ	the threshold to terminate the iteration

indicate the set of observed and unobserved user-item data, respectively. Further, we denote $\mathbf{I}^{\mathcal{O}}$ and $\mathbf{I}^{\mathcal{U}}$ as the indicator matrices for \mathcal{O} and \mathcal{U} , respectively. Similar to Matlab, we denote the i^{th} row of matrix \mathbf{R} as $\mathbf{R}(i, :)$, and the transpose of a matrix with a prime (i.e., $\mathbf{R}' \equiv \mathbf{R}^T$).

With these notations, we first define the basic OCCF problem.

PROBLEM 1. Basic OCCF Problem [24]

Given: (1) an $m \times n$ implicit, one-class feedback matrix \mathbf{R} , (2) a user u where $1 \leq u \leq m$, and (3) an item i where $1 \leq i \leq n$;

Find: the estimated preference of user u on item i .

In Problem 1, to estimate the preference of a given user on a given item, the only input needed is the feedback matrix \mathbf{R} . In addition to this feedback matrix, there might exist side information (e.g., the social relationships between users, the item similarity, etc.) that can be exploited in many applications. In this work, we focus on the pair-wise link information from both users and items. For example, a link between two users may indicate their similarity, intimacy, or friendship, while a link between two items may indicate their similarity or category. We use two symmetric matrices/graphs \mathbf{M} and \mathbf{N} to denote all the links between users and items, respectively. With these additional notations, we can define the following OCCF problem.

PROBLEM 2. OCCF Problem with Side Information

Given: (1) an $m \times n$ implicit, one-class feedback matrix \mathbf{R} , (2) an $m \times m$ user-user graph \mathbf{M} , (3) an $n \times n$ item-item graph \mathbf{N} , (4) a user u where $1 \leq u \leq m$, and (5) an item i where $1 \leq i \leq n$;

Find: the estimated preference of user u on item i .

2.2 Preliminaries

Before presenting our proposed solutions in the next two sections, let us first briefly review two representative OCCF methods.

The first one is the weighting-based method, which formulates the following minimization problem [24]

$$\min_{\mathbf{F}, \mathbf{G} \geq 0} \sum_{(u,i)} \mathbf{W}(u, i) (\mathbf{R}(u, i) - \mathbf{F}(u, :) \mathbf{G}(i, :)^T)^2 + \lambda_r (\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2) \quad (1)$$

where $\mathbf{W}(u, i)$ indicates the weight of the corresponding example, \mathbf{F} and \mathbf{G} are the non-negative low-rank approximations of \mathbf{R} , and λ_r is used to control the regularization of $\|\mathbf{F}\|_F^2$ and $\|\mathbf{G}\|_F^2$. Typically, a smaller weight is assigned for those unobserved examples. That is, $\mathbf{W}(u, i) = 1$ if $\mathbf{R}(u, i)$ is observed, and $\mathbf{W}(u, i) \in [0, 1]$ if $\mathbf{R}(u, i) = 0$. There are three basic strategies to assign $\mathbf{W}(u, i)$ [24]: uniform, user oriented, and item oriented. For simplicity, we adopt the uniform strategy, i.e., we set $\mathbf{W}(u, i) = \sqrt{w}$ for each unobserved $\mathbf{R}(u, i)$ where \sqrt{w} is the global weight. We will refer to this approach as *wZAN* in this paper.

Another approach is imputation based. The intuition behind this method is that the unobserved data in OCCF setting may contain many missing positive examples; therefore, we can impute a value for the unobserved example to indicate the possible feedback that the user would give to the item. Formally, the imputation-based method can be formulated as

$$\min_{\mathbf{F}, \mathbf{G} \geq 0} \sum_{(u,i)} ((\mathbf{R}(u, i) + \mathbf{P}(u, i)) - \mathbf{F}(u, :) \mathbf{G}(i, :)^T)^2 + \lambda_r (\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2) \quad (2)$$

where \mathbf{P} stands for the imputation matrix for the unobserved data. Typically, $\mathbf{P}(u, i) = 0$ if $\mathbf{R}(u, i) = 1$, and $\mathbf{P}(u, i) \in [0, 1]$ if $\mathbf{R}(u, i) = 0$. To fill in \mathbf{P} , we can resort to similar strategies as those for *wZAN*. For example, we can fill in $\mathbf{P}(u, i)$ uniformly with a global value

$p \in [0, 1]$ for the unobserved data. We can also adopt the user-oriented strategy or the item-oriented strategy. In this paper, we focus on the uniform strategy, and will refer to this approach as *iZAN* in the following. Note that the similar idea was also implicitly explored in [31].

3. THE PROPOSED FORMULATION

In this section, we present our formulations for the basic OCCF problem (i.e., Problem 1) and the OCCF with dual side information problem (i.e., Problem 2).

3.1 The Proposed Formulation for Problem 1

For Problem 1, the main challenge is from the ambiguity of the unobserved data, i.e., the unobserved data is not necessarily negative examples but a mixing of negative examples and missing positive examples. To deal with the ambiguity problem, the weighting-based method (e.g., *wZAN* in Eq. (1)) adds a weight for unobserved examples, while the imputation-based method (e.g., *iZAN* in Eq. (2)) imputes the unobserved data. The key observation here is that these two methods are complementary to each other: we are more certain about the observed data, and we have a relatively lower confidence of the imputed data. This leads to a natural choice of integrating these two methods by putting a smaller weight on the contribution of the imputed data, i.e.,

$$\min_{\mathbf{F}, \mathbf{G} \geq 0} \sum_{(u,i)} \mathbf{W}(u,i) (\mathbf{R}(u,i) + \mathbf{P}(u,i) - \mathbf{F}(u,:) \mathbf{G}(i,:))'^2 + \lambda_r (\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2) \quad (3)$$

where $\mathbf{W}(u,i)$ and $\mathbf{P}(u,i)$ are defined in Eq. (1) and Eq. (2), respectively. In this formulation, we introduce the imputed value $\mathbf{P}(u,i)$ of an unobserved example to indicate the likelihood that the user would favor the corresponding item. In the meanwhile, we introduce a small weight $\mathbf{W}(u,i)$ to indicate its lower reliability. We will refer to this approach as *wiZAN*.

Scalability issues. As we can see from Eq. (3), *wiZAN* optimizes over a dense matrix $(\mathbf{R} + \mathbf{P})$ of size $m \times n$, and introduces another $m \times n$ matrix \mathbf{W} . Directly optimizing Eq. (3) might be computationally prohibitive in terms of both time and space in many large-scale applications. We will propose scalable algorithms to tackle this issue in the next section.

3.2 The Proposed Formulation for Problem 2

For Problem 2, we include both the user-user graph and the item-item graph as additional inputs. Effectively leveraging such side information might not only help alleviate the cold-start problem, but also improve the overall performance for recommender systems. Here, we describe how we incorporate the side information. We take the user-user side information as an example, but similar treatment can be applied on the item-item side information.

For user side, the basic idea is to employ the Homophily effect [22], i.e., similar users tend to share similar preferences for items. In other words, if two users are connected (i.e., a non-zero and/or larger $\mathbf{M}(u,v)$), they might have similar latent preferences for items (i.e., $\mathbf{F}(u,:)$ and $\mathbf{F}(v,:)$ are close to each other). Therefore, for two users u and v , we add the following constraint on their preferences

$$\min \sum_{u=1}^m \sum_{v=1}^m \mathbf{M}(u,v) \|\mathbf{F}(u,:) - \mathbf{F}(v,:)\|_2^2 \quad (4)$$

where m is the total number of users, and \mathbf{M} is the $m \times m$ user-user graph for these users. As we can see from Eq. (4), the link $\mathbf{M}(u,v)$ would encourage the corresponding latent preferences (i.e., $\mathbf{F}(u,:)$

and $\mathbf{F}(v,:)$ to be close to each other. Notice that it has no penalty for two dis-connected users (i.e., $\mathbf{M}(u,v) = 0$).

We can further formulate the constraint in Eq. (4) as a graph regularization term with the following equations

$$\begin{aligned} & \frac{1}{2} \sum_{u=1}^m \sum_{v=1}^m \mathbf{M}(u,v) \|\mathbf{F}(u,:) - \mathbf{F}(v,:)\|_2^2 \\ &= \frac{1}{2} \sum_{u=1}^m \sum_{v=1}^m \sum_{k=1}^r \mathbf{M}(u,v) (\mathbf{F}(u,k) - \mathbf{F}(v,k))^2 \\ &= \sum_{u=1}^m \sum_{v=1}^m \sum_{k=1}^r \mathbf{M}(u,v) \mathbf{F}^2(u,k) - \sum_{u=1}^m \sum_{v=1}^m \sum_{k=1}^r \mathbf{M}(u,v) \mathbf{F}(u,k) \mathbf{F}(v,k) \\ &= \sum_{k=1}^r \mathbf{F}(:,k)' (\mathbf{D}_M - \mathbf{M}) \mathbf{F}(:,k) \\ &= \text{tr}(\mathbf{F}' (\mathbf{D}_M - \mathbf{M}) \mathbf{F}) \end{aligned} \quad (5)$$

where $\text{tr}(\cdot)$ stands for the matrix trace, r is the rank of \mathbf{F} , and \mathbf{D}_M is the degree matrix for \mathbf{M} with $\mathbf{D}_M(u,u) = \sum_{v=1}^m \mathbf{M}(u,v)$.

Similarly, we have the item-side regularization term $\text{tr}(\mathbf{G}' (\mathbf{D}_N - \mathbf{N}) \mathbf{G})$, where \mathbf{N} is the symmetric item-item graph which represents item-item similarity, and \mathbf{D}_N is the degree matrix for \mathbf{N} .

Finally, we incorporate the two regularization terms into Eq. (3), resulting in our dual-regularized OCCF model for Problem 2

$$\min_{\mathbf{F}, \mathbf{G} \geq 0} \sum_{(u,i)} \underbrace{\mathbf{W}(u,i)}_{\text{weights/reliability}} (\underbrace{\mathbf{R}(u,i) + \mathbf{P}(u,i)}_{\text{imputed values/likelihood}} - \mathbf{F}(u,:) \mathbf{G}(i,:))'^2 + \lambda_r (\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2) + \lambda_F \underbrace{\text{tr}(\mathbf{F}' (\mathbf{D}_M - \mathbf{M}) \mathbf{F})}_{\text{user homophily}} + \lambda_G \underbrace{\text{tr}(\mathbf{G}' (\mathbf{D}_N - \mathbf{N}) \mathbf{G})}_{\text{item homophily}} \quad (6)$$

where λ_F and λ_G are used to control the importance of the two regularization terms. Notice that the two regularization terms, $(\mathbf{D}_M - \mathbf{M})$ and $(\mathbf{D}_N - \mathbf{N})$ are the graph Laplacian of the user-user graph and item-item graph, respectively. Actually, such a dual regularization can be plugged into many existing methods (e.g., *wZAN* and *iZAN*), as we will show in the experimental section.

With the \mathbf{F} and \mathbf{G} matrices derived from the above formulations (e.g., Eq. (6) and Eq. (3)), we can estimate the preference of user u on item i as $\hat{\mathbf{R}}(u,i) = \mathbf{F}(u,:) \mathbf{G}(i,:)$.

4. THE PROPOSED ALGORITHM

In this section, we present the algorithm (*wiZAN-Dual*) to solve the OCCF problem with side information in Eq. (6), followed by some effectiveness and efficiency analysis. The algorithm (*wiZAN*) for Eq. (3) can be derived from *wiZAN-Dual* by ignoring the dual regularization terms.

4.1 The *wiZAN-Dual* Algorithm

Unfortunately, the optimization problem in Eq. (6) is not jointly convex due to the coupling between \mathbf{F} and \mathbf{G} . Therefore, instead of seeking for a global optimal solution, we aim to find a local minimum by alternatively updating \mathbf{F} and \mathbf{G} while fixing the other. Next, we show how to update \mathbf{F} when \mathbf{G} is fixed. The update of \mathbf{G} can be done in a similar way.

When \mathbf{G} is fixed, the optimization problem in Eq. (6) becomes the minimization problem of the following equation (by dropping some constant terms) wrt the matrix \mathbf{F}

$$\begin{aligned} J &= \|\mathbf{W} \odot ((\mathbf{R} + \mathbf{P}) - \mathbf{F} \mathbf{G}')\|_F^2 + \lambda_r \|\mathbf{F}\|_F^2 + \lambda_F \text{tr}(\mathbf{F}' (\mathbf{D}_M - \mathbf{M}) \mathbf{F}) \\ \text{s.t. } & \mathbf{F} \geq 0 \end{aligned} \quad (7)$$

where \odot is the Hadamard product with $[\mathbf{A} \odot \mathbf{B}](u,i) = \mathbf{A}(u,i) \mathbf{B}(u,i)$ for any two matrices with the same size.

Since both the user-user graph \mathbf{M} and its degree matrix \mathbf{D}_M are symmetric, the derivative of J wrt \mathbf{F} can be computed as

$$\frac{1}{2} \frac{\partial J}{\partial \mathbf{F}} = -(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + (\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} + \lambda_r \mathbf{F} + \lambda_F \mathbf{D}_M \mathbf{F} - \lambda_F \mathbf{M}\mathbf{F} \quad (8)$$

In Eq. (8), \mathbf{M} is a sparse non-negative matrix, and \mathbf{D}_M is a diagonal non-negative matrix. A fixed-point solution of Eq. (8) with the non-negativity constraint leads to the following multiplicative updating rule for \mathbf{F}

$$\mathbf{F}(u, k) \leftarrow \mathbf{F}(u, k) \sqrt{\frac{[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + \lambda_F \mathbf{M}\mathbf{F}](u, k)}{[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} + \lambda_r \mathbf{F} + \lambda_F \mathbf{D}_M \mathbf{F}](u, k)}} \quad (9)$$

The $(\mathbf{R} + \mathbf{P})$ matrix in Eq. (9) is extremely large in many recommender systems, causing severe scalability issues in terms of both time and storage. To tackle this issue, we propose an efficient algorithm to scale up the updating process in terms of both time and storage. Before presenting our algorithm, we need to further define matrix $\tilde{\mathbf{R}}_1$. We denote $\tilde{\mathbf{R}}_1$ as the sparse matrix whose elements are predicted by \mathbf{F} and \mathbf{G} on the observed examples in \mathbf{R} . That is,

$$\tilde{\mathbf{R}}_1(u, i) = \begin{cases} \mathbf{F}(u, :) \mathbf{G}(i, :) & \text{if } (u, i) \in \mathcal{O} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Based on the $\tilde{\mathbf{R}}_1$ matrix, we present the new updating rule for \mathbf{F} as follows

$$\mathbf{F}(u, k) \leftarrow \mathbf{F}(u, k) \sqrt{\frac{\mathbf{A}_1(u, k)}{\mathbf{B}_1(u, k)}} \quad (11)$$

where \mathbf{A}_1 and \mathbf{B}_1 are defined as

$$\begin{aligned} \mathbf{A}_1 &= (1 - wp)\mathbf{R}\mathbf{G} + wp\mathbf{1}_{m \times 1}(\mathbf{1}_{1 \times n}\mathbf{G}) + \lambda_F \mathbf{M}\mathbf{F} \\ \mathbf{B}_1 &= (1 - w)\tilde{\mathbf{R}}_1 \mathbf{G} + w\mathbf{F}(\mathbf{G}'\mathbf{G}) + \lambda_r \mathbf{F} + \lambda_F \mathbf{D}_M \mathbf{F} \end{aligned} \quad (12)$$

Here, \sqrt{w} is the global weight assigned to the unobserved examples, p is the global imputation value, and $\mathbf{1}_{1 \times n}$ is a $1 \times n$ vector with all 1s.

Following similar steps, we can have the updating rule for \mathbf{G}

$$\mathbf{G}(i, k) \leftarrow \mathbf{G}(i, k) \sqrt{\frac{\mathbf{A}_2(i, k)}{\mathbf{B}_2(i, k)}} \quad (13)$$

with \mathbf{A}_2 and \mathbf{B}_2 in the following form

$$\begin{aligned} \mathbf{A}_2 &= (1 - wp)\mathbf{R}'\mathbf{F} + wp\mathbf{1}_{n \times 1}(\mathbf{1}_{1 \times m}\mathbf{F}) + \lambda_G \mathbf{N}\mathbf{G} \\ \mathbf{B}_2 &= (1 - w)\tilde{\mathbf{R}}_1' \mathbf{F} + w\mathbf{G}(\mathbf{F}'\mathbf{F}) + \lambda_r \mathbf{G} + \lambda_G \mathbf{D}_N \mathbf{G} \end{aligned} \quad (14)$$

where \mathbf{N} is a sparse, non-negative matrix containing the item links, and \mathbf{D}_N is the diagonal degree matrix for \mathbf{N} .

Finally, we summarize the overall algorithm for solving Eq. (6) in Alg. 1. As we can see from the algorithm, after we initialize the \mathbf{F} and \mathbf{G} matrices (Step 1), the algorithm begins the iteration procedure. In each iteration, the algorithm first computes the $\tilde{\mathbf{R}}_1$ matrix (Step 3), and then alternatively updates \mathbf{F} and \mathbf{G} (Steps 4-9 and Steps 10-15, respectively). We use the following criteria to terminate the iteration procedure: either the Frobenius norm between successive estimates of both \mathbf{F} and \mathbf{G} is below our threshold ξ or the maximum iteration step l is reached. Finally, we can predict the preference of user u on item i by $\mathbf{F}(u, :) \mathbf{G}(i, :)$.

4.2 Algorithm Analysis

Here, we briefly analyze the optimality, convergency, and computational complexity of our algorithm.

Algorithm 1 The *wiZAN-Dual* Algorithm.

Input: \mathbf{R} , \mathbf{M} , \mathbf{N} , rank r , global weight w , and global imputation value p

Output: \mathbf{F} and \mathbf{G}

```

1: initialize  $\mathbf{F}$  and  $\mathbf{G}$  randomly;
2: while not convergent do
3:   compute  $\tilde{\mathbf{R}}_1$  as defined in Eq. (10);
4:   compute  $\mathbf{A}_1$  and  $\mathbf{B}_1$  as defined in Eq. (12);
5:   for  $u = 1 : m$  do
6:     for  $k = 1 : r$  do
7:       update  $\mathbf{F}(u, k)$  as defined in Eq. (11);
8:     end for
9:   end for
10:  compute  $\mathbf{A}_2$  and  $\mathbf{B}_2$  as defined in Eq. (14);
11:  for  $i = 1 : n$  do
12:    for  $k = 1 : r$  do
13:      update  $\mathbf{G}(i, k)$  as defined in Eq. (13);
14:    end for
15:  end for
16: end while
17: return  $\mathbf{F}$  and  $\mathbf{G}$ ;

```

We first show the correctness of Eq. (11) for updating \mathbf{F} , by proving that the fixed-point solution of Eq. (11) satisfies the KKT condition. The correctness of Eq. (13) for updating \mathbf{G} can be proved analogously.

THEOREM 1. Correctness of Eq. (11). *The fixed-point solution of Eq. (11) satisfies the KKT condition.*

PROOF. In order to prove the theorem, we will first show that Eq. (9) satisfies the KKT condition, and then show the equivalence between Eq. (9) and Eq. (11). We start with Lagrangian function of Eq. (7)

$$\begin{aligned} L_J &= \|\mathbf{W} \odot ((\mathbf{R} + \mathbf{P}) - \mathbf{F}\mathbf{G}')\|_F^2 + \lambda_r \|\mathbf{F}\|_F^2 + \lambda_F \text{tr}(\mathbf{F}'\mathbf{D}_M \mathbf{F}) \\ &\quad - \lambda_F \text{tr}(\mathbf{F}'\mathbf{M}\mathbf{F}) - \text{tr}(\Lambda'\mathbf{F}) \end{aligned}$$

where Λ is the Lagrange multiplier. Let the derivative of the above equation L_J equal to 0, we have

$$\begin{aligned} 2(-(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + (\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} \\ + \lambda_r \mathbf{F} + \lambda_F \mathbf{D}_M \mathbf{F} - \lambda_F \mathbf{M}\mathbf{F}) = \Lambda \end{aligned}$$

From the KKT complementary slackness condition, we have

$$\begin{aligned} [-(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + (\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} \\ + \lambda_r \mathbf{F} + \lambda_F \mathbf{D}_M \mathbf{F} - \lambda_F \mathbf{M}\mathbf{F}](u, k) \mathbf{F}(u, k) = 0 \end{aligned}$$

Clearly, a fixed point of the updating rule in Eq. (9) satisfies the above equation.

Next, we show the equivalence between Eq. (9) and Eq. (11). We first introduce several notations. We denote $\tilde{\mathbf{R}}_2$ as the matrix whose values are predicted by \mathbf{F} and \mathbf{G} on the unobserved examples in \mathbf{R} , i.e., $\tilde{\mathbf{R}}_1 + \tilde{\mathbf{R}}_2 = \mathbf{F}\mathbf{G}'$. We use \mathbf{I}^O and \mathbf{I}^U as the indicator matrices for observed and missing data, respectively. The full indicator matrix is $\mathbf{I}^A = \mathbf{I}^O + \mathbf{I}^U$. Then, we have the following two equations:

$$\begin{aligned} &(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} \\ &= (\mathbf{I}^O \odot \mathbf{R} + (w\mathbf{I}^U) \odot \mathbf{P})\mathbf{G} \\ &= (\mathbf{R} - (w\mathbf{I}^O) \odot (p\mathbf{I}^O) + (w\mathbf{I}^A) \odot (\mathbf{P} + (p\mathbf{I}^O)))\mathbf{G} \\ &= (1 - wp)\mathbf{R}\mathbf{G} + wp\mathbf{1}_{m \times 1}(\mathbf{1}_{1 \times n}\mathbf{G}) \end{aligned}$$

and

$$\begin{aligned}
& (\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} \\
= & (\mathbf{I}^O \odot \tilde{\mathbf{R}}_1 + (w\mathbf{I}^U) \odot \tilde{\mathbf{R}}_2)\mathbf{G} \\
= & (\tilde{\mathbf{R}}_1 - (w\mathbf{I}^O) \odot \tilde{\mathbf{R}}_1 + (w\mathbf{I}^U) \odot (\mathbf{F}\mathbf{G}'))\mathbf{G} \\
= & (1 - w)\tilde{\mathbf{R}}_1\mathbf{G} + w\mathbf{F}(\mathbf{G}'\mathbf{G})
\end{aligned}$$

Finally, we can get the updating rule in Eq. (11) by substituting the above two equations into Eq. (9), which completes the proof. \square

In Theorem 1, we have shown that the updating rule in Eq. (11) yields a correct solution for minimizing Eq. (7) at convergence. Next, we prove that the updating rule in Eq. (11) is guaranteed to converge.

THEOREM 2. Convergence of Eq. (11). *Under the updating rule of Eq. (11), Eq. (7) decreases monotonically.*

PROOF. See the appendix. \square

Combining Theorem 1 and Theorem 2 together, we can have the following corollary, which states that Alg. 1 finds a local optimum for Eq. (6). Given that the original optimization problem in Eq. (6) is not jointly convex wrt \mathbf{F} and \mathbf{G} , such a local minimum is acceptable in practice.

COROLLARY 1. Effectiveness of Alg. 1. *Alg. 1 finds a local minimum for the optimization problem in Eq. (6).*

PROOF. Omitted for brevity. \square

The time complexity and space complexity of the proposed algorithm are summarized in the following lemmas, which basically state that Alg. 1 scales linearly wrt the total number observed examples (i.e., $|\mathbf{R}| + |\mathbf{M}| + |\mathbf{N}|$) and the total number of users and items (i.e., $m+n$) in both time and space. Notice that all the three matrices (\mathbf{R} , \mathbf{M} , and \mathbf{N}) are often very sparse (e.g., $|\mathbf{R}| \ll mn$, $|\mathbf{M}| \ll m^2$, etc). In contrast, if we directly use the updating rule in Eq. (9), it would cost us quadratic complexity (e.g., $O(mn)$) in both time and space.

LEMMA 1. Time complexity of Alg. 1. *The time complexity of Alg. 1 is $O((|\mathbf{R}| + |\mathbf{M}| + |\mathbf{N}|)rl + (m+n)r^2l)$.*

PROOF. Omitted for brevity. \square

LEMMA 2. Space complexity of Alg. 1. *The space complexity of Alg. 1 is $O(|\mathbf{R}| + |\mathbf{M}| + |\mathbf{N}| + (m+n)r)$.*

PROOF. Omitted for brevity. \square

5. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluations. The experiments are designed to answer the following questions.

- *Effectiveness*: How accurate are the proposed algorithms for one-class collaborative filtering? How do they perform for the cold-start users/items?
- *Efficiency*: How scalable are the proposed algorithms?

5.1 Experimental Setup

5.1.1 Data Sets Description

We use two real data sets: *Ciao* and *Epinions* [33, 35]. For both data sets, we randomly select 50% ratings as the training set and use the rest as the test set. For one-class experiments, we remove all the ratings that are no greater than 3, and relabel ratings 4 and

Table 2: The statistics of *Ciao* and *Epinions* data sets.

Data	<i>Ciao</i>	<i>Epinions</i>
# of users	6,102	33,725
# of items	12,082	43,542
# of positive examples	117,731	500,478
# of user links	151,722	656,910
# of item links	283,284	498,794
sparsity of positive examples	0.160%	0.034%
sparsity of user links	0.407%	0.058%
sparsity of item links	0.194%	0.026%

5 as 1 (positive examples). For the user-side regularization, we use the trust relationships between users. In particular, we assign a trust link $\mathbf{M}_{uv} = 1$ if either u trusts v or v trusts u . The resulting \mathbf{M} matrix is sparse (see Table 2). For the item-side regularization, we aggregate the reviews as a document for each item, and then compute the cosine similarity between the TF-IDF vectors of these documents. To keep the item-item graph sparse, we assign $\mathbf{N}_{ij} = 1$ if the cosine similarity between item i and item j is larger than 0.4. The statistics of the two data sets are summarized in Table 2.

5.1.2 Compared Methods

First, we compare with three existing methods for explicit, multi-class collaborative filtering. Notice that both *GWNMF* and *SR* also formulate the user and item side information as graph regularization terms. Despite their own success in the multi-class case, their algorithms would lead to trivial solutions in the one-class case (with the two latent matrices \mathbf{F} and \mathbf{G} being all $1/\sqrt{r}$). As we will show soon, they inevitably result in poor recommendation performance in the one-class case.

- *ZAM*. *ZAM* is a baseline method for explicit, multi-class collaborative filtering. It treats all zeros as missing values, i.e., $\mathbf{W}(u, i) = 1$ for the observed (positive) examples and $\mathbf{W}(u, i) = 0$ for the unobserved examples in Eq. (1).
- *GWNMF* [9]. *GWNMF* is proposed to combine the merits of memory-based method with neighborhood information. Similar to our method, *GWNMF* constructs the user and item side information as two similarity graphs, and adds such graphs as regularization terms into the objective function.
- *SR* [19]. On the methodology level, *SR* is similar to *GWNMF* except that *SR* exploits both similar users/items and dissimilar users/items, and *SR* does not add the non-negativity constraint while *GWNMF* does.

We also compare the following algorithms for one-class setting:

- *ZAN*. *ZAN* is a baseline method, and it treats all zeros as negative examples, i.e., $\mathbf{W}(u, i) = 1$ for both observed and unobserved examples in Eq. (1).
- *wZAN* [24, 23]. *wZAN* weights the contribution of the unobserved data by setting $\mathbf{W}(u, i) = \sqrt{w}$ for the unobserved examples where $\sqrt{w} \in [0, 1]$ is the global weight (Eq. (1)).
- *iZAN*. *iZAN* stands for the imputation-based method as shown in Eq. (2). This can be seen as a special case of the proposed *wiZAN-Dual* method if we ignore the weighting matrix and the dual regularization terms.
- *ldNMF* [31]. *ldNMF* borrows the idea of low-density methods from semi-supervised SVM [4]. In particular, it treats the unobserved entries as optimization variables.

- *RG* [25]. *RG* is a sampling-based method proposed for one-class recommendation. The basic idea behind *RG* is to generate a random graph that preserves the degree distributions of the observed examples. The generated graph includes all the positive examples and a set of negative examples.
- *MSCMF* [38]. *MSCMF* is proposed for predicting drug-target interactions. It adds constraints from user/item neighborhood, and it can deal with multiple similarity matrices on both sides.
- *wiZAN*. *wiZAN* is the proposed method for the basic OCCF problem when side information is not available (Eq. (3)). It is a special case of the *wiZAN-Dual* method if we ignore dual regularization.
- *wiZAN-Dual*. *wiZAN-Dual* is the proposed method that integrates the imputation-based method, the weighting-based method, and the dual regularization from both users and items (Eq. (6)).

For the results reported in this section, we use the same initializations of \mathbf{F} and \mathbf{G} , and we fix the global weight $\sqrt{w} = 0.1$, the global imputation value $p = 0.01$, and rank $r = 10$ unless otherwise stated. For other parameters, we set maximum iteration $l = 100$, the termination threshold $\xi = 10^{-6}$, and the regularization parameters $\lambda_r = 0.1$, $\lambda_F = 1$, $\lambda_G = 0.1$.

5.1.3 Evaluation Metrics

To evaluate the effectiveness of the compared methods, we adopt three widely used evaluation metrics for OCCF.

The first metric is *Half-Life Utility (HLU)* [3, 24]. *HLU* estimates how likely a user will view/choose an item from a ranked list, with the assumption that the user will view each consecutive item in the list with an exponential decay of possibility. A larger *HLU* indicates better recommendation performance.

The second metric is *Mean Average Precision (MAP)* [18]. *MAP* measures the overall performance based on precision at different recall levels. It calculates the mean of the average precision (AP) over all users in the test set. A larger *MAP* indicates better recommendation performance.

The third metric is a recall-oriented metric *Mean Percentage Ranking (MPR)* [12]. *MPR* measures the user satisfaction of items in a ranked list. It is expected that a randomly produced list would have a *MPR* of 50%. A smaller *MPR* indicates better recommendation performance.

For efficiency experiments, we simply report the wall-clock time of the proposed algorithms. All the experiments were run on a machine with eight 3.4GHz Intel Cores and 24GB memory.

5.2 Effectiveness Results

(A) *Effectiveness Comparisons*. We first compare the overall effectiveness performance of the proposed methods with that of the existing methods. The results on *Ciao* data and *Epinions* data are shown in Table 3 and Table 4, respectively. Larger *HLU/MAP* and smaller *MPR* are better. *ldNMF* is computationally prohibitive on *Epinions* data due to its quadratic complexity.

There are several observations from the tables. First of all, the proposed *wiZAN-Dual* outperforms all the compared methods in all evaluation metrics on both data sets. For example, on the *Ciao* data, *wiZAN-Dual* outperforms the best existing competitors by 15.0%, 21.1%, and 10.1% wrt *HLU*, *MAP*, and *MPR*, respectively; on the *Epinions* data, *wiZAN-Dual* outperforms the best existing competitors by 9.9%, 17.3%, and 7.9% wrt *HLU*, *MAP*, and *MPR*, respectively. Second, as expected, the first three methods (*ZAM*,

Table 3: Effectiveness results on *Ciao* data. Larger *HLU/MAP* and smaller *MPR* are better. *wiZAN-Dual* significantly outperforms all the compared methods.

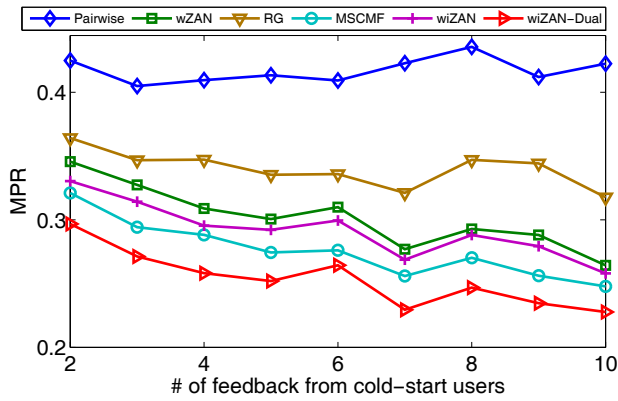
Methods	<i>HLU</i>	<i>MAP</i>	<i>MPR</i>
<i>ZAM</i>	0.4876	0.0041	0.3744
<i>GWNMF</i>	0.3453	0.0053	0.3410
<i>SR</i>	0.0656	0.0011	0.5910
<i>ZAN</i>	5.1092	0.0228	0.3106
<i>wZAN</i>	5.9981	0.0275	0.2846
<i>iZAN</i>	5.5198	0.0248	0.3106
<i>ldNMF</i>	5.5190	0.0248	0.3227
<i>RG</i>	5.6723	0.0254	0.3478
<i>MSCMF</i>	5.5799	0.0272	0.2639
<i>wiZAN</i>	6.3621	0.0297	0.2806
<i>wiZAN-Dual</i>	6.8990	0.0333	0.2373

Table 4: Effectiveness results on *Epinions* data. Larger *HLU/MAP* and smaller *MPR* are better. *wiZAN-Dual* significantly outperforms all the compared methods. The *ldNMF* method is computationally prohibitive on *Epinions* data.

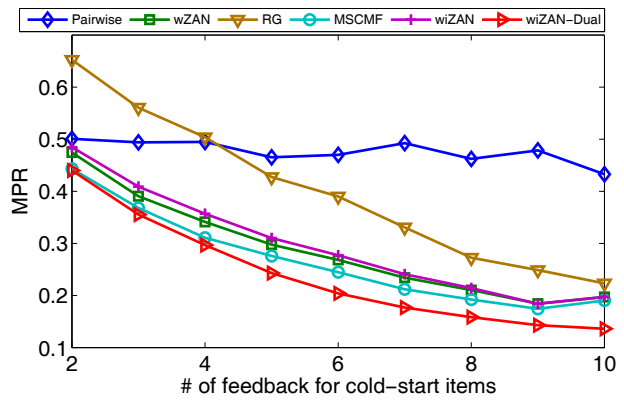
Methods	<i>HLU</i>	<i>MAP</i>	<i>MPR</i>
<i>ZAM</i>	0.1349	0.0014	0.3036
<i>GWNMF</i>	0.0254	0.0012	0.2643
<i>SR</i>	0.0088	0.0002	0.6562
<i>ZAN</i>	2.6358	0.0123	0.2308
<i>wZAN</i>	2.7367	0.0139	0.2179
<i>iZAN</i>	3.3401	0.0173	0.3233
<i>ldNMF</i>	-	-	-
<i>RG</i>	2.0062	0.0129	0.3032
<i>MSCMF</i>	2.7501	0.0140	0.2067
<i>wiZAN</i>	3.5409	0.0198	0.2190
<i>wiZAN-Dual</i>	3.6700	0.0203	0.1903

GWNMF, and *SR*) that are designed for the MCCF setting perform poorly in the OCCF setting. Third, compared to the other methods that are proposed for the OCCF setting, our *wiZAN* can already achieve better or close performance. For example, on both data sets, *wiZAN* is better than all the existing competitors in all three evaluation metrics except the *MPR* metric of the *MSCMF* method. Recall that the *MSCMF* method also uses the side information. This indicates that *wiZAN* outperforms all the compared methods for the basic OCCF problem defined in Problem 1. Forth, the overall performance of all the methods on *Ciao* data is better than that on the *Epinions* data. This is probably due to the fact that the *Epinions* data is much sparser than the *Ciao* data (e.g., 0.034% sparsity of *Epinions* data vs. 0.160% sparsity of *Ciao* data).

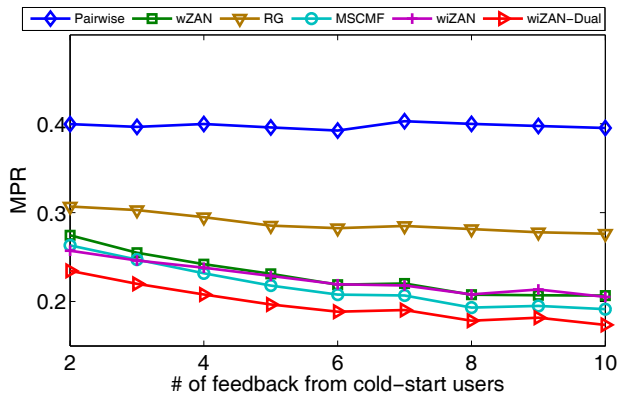
(B) *Effectiveness of Dual Regularization*. The proposed dual regularization is applicable to many existing methods. We have already shown the usefulness of dual regularization by incorporating them into our own *wiZAN* as shown in Table 3 and Table 4. Next, we further verify the effectiveness of the dual regularization terms by adding them into *wZAN* and *iZAN*. The results on the two data sets are shown in Table 5 and Table 6, respectively. As we can see, both *wZAN-Dual* and *iZAN-Dual* perform better than the corresponding cases when the regularization terms are not added. For example, *wZAN-Dual* improves *wZAN* by 13.4%, 18.9%, and 14.1% wrt *HLU*, *MAP*, and *MPR* on the *Ciao* data, respectively. This result indicates that dual regularization indeed helps in OCCF recommendation.



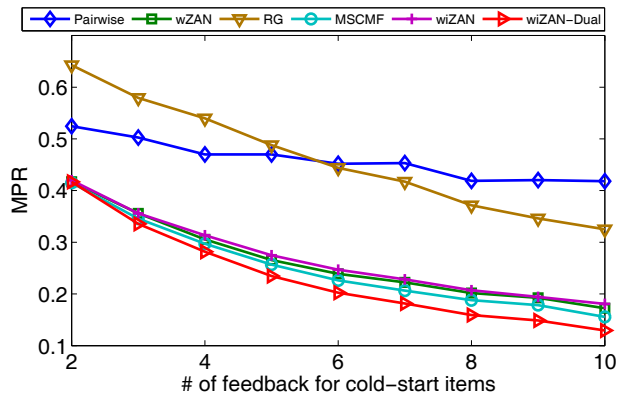
(a) Cold-start users on *Ciao* data



(b) Cold-start items on *Ciao* data



(c) Cold-start users on *Epinions* data



(d) Cold-start items on *Epinions* data

Figure 1: The effectiveness comparisons in cold-start scenarios. Smaller MPR is better. $wiZAN-Dual$ outperforms all the compared methods for both cold-start users and cold-start items on both data sets.

Table 5: Effectiveness of side information on *Ciao* data. Larger HLU/MAP and smaller MPR are better. Our dual regularization improves the prediction accuracy in all cases.

Methods	HLU	MAP	MPR
wZAN	5.9981	0.0275	0.2846
wZAN-Dual	6.8044	0.0327	0.2444
iZAN	5.5198	0.0248	0.3106
iZAN-Dual	6.4389	0.0293	0.2927

(C) *Effectiveness in Cold-start Scenarios.* Next, we put our focus on the cold-start scenarios. As mentioned in introduction, one of the advantages of our method is to alleviate the cold-start problem in recommender systems. Here, we compare the effectiveness of $wiZAN-Dual$ in the cold-start scenarios with several best competitors including $wiZAN$, $MSCMF$, RG , and $wZAN$. We also compare with an additional $Pairwise$ method [26]. The $Pairwise$ method is specially designed for cold-start users/items, and it requires the user features and item features as input. To apply the $Pairwise$ method to our problem setting, we perform a spectral decomposition method to translate \mathbf{M} and \mathbf{N} to the feature representation for users/items. The results are shown in Fig 1, where x-axis indicates the number of positive feedback given by the cold-start users or received by the cold-start items in the training set, and y-axis indicates the MPR metric.

Table 6: Effectiveness of side information on *Epinions* data. Larger HLU/MAP and smaller MPR are better. Our dual regularization improves the prediction accuracy in all cases.

Methods	HLU	MAP	MPR
wZAN	2.7367	0.0139	0.2179
wZAN-Dual	3.1902	0.0166	0.1984
iZAN	3.3401	0.0173	0.3233
iZAN-Dual	3.4833	0.0179	0.2889

As we can see from the figures, $wiZAN-Dual$ outperforms all the compared methods for both cold-start users and cold-start items on both data sets. Specially, $wiZAN-Dual$ is better than $wiZAN$, which directly indicates the importance of dual regularization in the cold-start scenarios. The $MSCMF$ method performs the second best in the compared methods. The reason is that although in a different way, $MSCMF$ also considers the side information from both users and items.

5.3 Efficiency Results

Finally, we evaluate the efficiency of $wiZAN-Dual$ by reporting the wall-clock time of the training stage (i.e., Alg. 1). We use the subsets of the data sets to test the scalability of the proposed algorithm. The results are shown in Fig. 2. As we can see from the figures, our algorithm scales linearly wrt the total number of observations (i.e., $|\mathbf{R}| + |\mathbf{M}| + |\mathbf{N}|$) and the total number of users and items

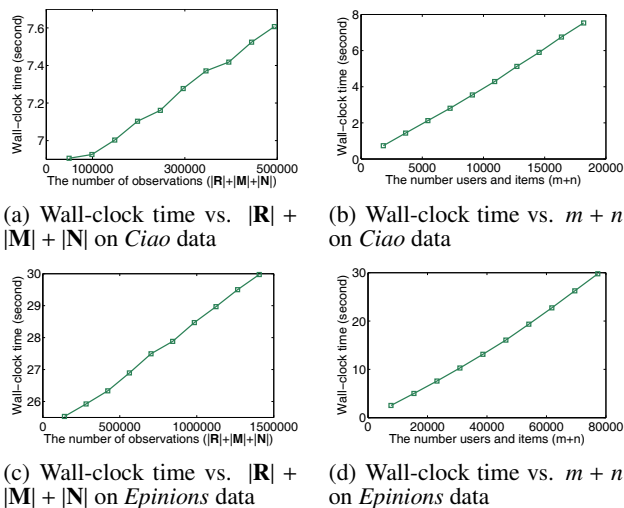


Figure 2: Scalability of the proposed algorithm. It scales linearly wrt the data size.

(i.e., $m + n$), which is consistent with our analysis in Lemma 1. Additionally, the algorithm is very efficient, finishing the training stage within 30 seconds for both data sets.

6. RELATED WORK

In this section, we briefly review the related work including the existing methods for multi-class collaborative filtering and one-class collaborative filtering, and the existing solutions for the cold-start problem.

Multi-Class Collaborative Filtering. Most of the existing collaborative filtering methods are proposed for explicit, multi-class recommender systems. Typically, collaborative filtering methods are categorized into memory-based methods and model-based methods [13, 14]. A combination of memory-based method and model-based method has also been explored [15, 19].

Considering the wide existence of social links between users, many researchers begin to incorporate these social links into collaborative filtering [20, 21, 36, 30, 34]. The key idea of these methods is that the linked users tend to have similar latent preferences for items. By applying similar idea to the item side, Gu et al. [9] employ the link information from both user side and item side.

One-Class Collaborative Filtering. Although one-class collaborative filtering is less visited compared to the multi-class setting, it is widely applicable in many real situations. According to how they treat the unobserved data, existing solutions can be categorized into three classes: *weighting-based* methods and *sampling-based* methods [24, 23], as well as *imputation-based* methods. For example, Hu et al. [12] propose a weighting-based method for recommending TV shows, and they obtain the weight from the number of minutes that a given show was watched. Paquet and Koenigstein [25] propose a sampling-based method where the degree distributions of users/items are preserved. Sindhvani et al. [31] propose to treat the unobserved data as optimization variables, which is essentially the imputation-based method.

Side information is also exploited by several researchers. For example, Li et al. [18] propose to leverage the users' past queries to construct the user-item similarity, and use such similarity to improve the recommendation performance; Zheng et al. [38] propose to employ multiple similarity matrices between users/items for drug-target interaction prediction. Other related proposals for

one-class setting include the rank-based optimization objective [28], the matrix co-factorization method [8], the combination of sentiment analysis and neighborhood method [27], etc.

Cold-Start Problem. Cold-start problem is one of the key challenges in recommender systems. In cold-start scenarios, it is relatively difficult to provide accurate recommendations for cold-start users [39], cold-start items [1], or both [26]. Existing solutions for cold-start problem can be categorized into three classes: *interview based*, *adjustment based*, and *side-information based*.

In the interview-based methods, an additional set of items is usually provided in the sign-up phase to collect the preferences of the cold-start users [11, 39, 32]. For example, Zhou et al. [39] use decision tree to select the set of interview items; Harpale and Yang [11] identify the interview set by active learning. One problem of the interview-based methods is that they bring additional burdens to the cold-start users. The second class of adjustment-based methods mainly focus on how to make full use of the small amount of ratings from cold-start users or for cold-start items. For example, Hacker and Ahn [10] introduce an online game during which the preferences of cold-start users can be adjusted. Methods in the third class exploit the side information to alleviate the cold-start problem. In this class, existing methods can be further categorized into attribute-based methods and link-based methods according to the type of side information they used. For attribute-based methods, Schein et al. [29] use item attributes such as item content; Park et al. [26] and Zhang et al. [37] leverage both user attributes and item attributes such as the demographical information. In contrast to attribute-based methods, link-based methods mainly employ the the social relationships between users [20, 30]. Our method falls into this sub-category, and we encode the link-based side information from both users and items.

7. CONCLUSIONS

In this paper, we have proposed a unified model *wiZAN-Dual* for one-class collaborative filtering. In *wiZAN-Dual*, we address (1) the ambiguity challenge by imputing each unobserved user-item pair to indicate the probability that the user would prefer the item, together with a weight to control the contribution/reliability of the imputed data; and (2) the sparseness challenge by exploiting the relationships between users/items (i.e., the user/item side information). We propose efficient algorithms for *wiZAN-Dual*, and analyze our algorithms in terms of optimality, correctness, and complexity. Our experimental evaluations on real benchmark data sets show that the proposed method leads to significant improvement over the state-of-the-art methods in prediction accuracy, while enjoying the linear scalability in both time and space.

8. ACKNOWLEDGMENTS

This work is supported by the National 863 Program of China (No. 2012AA011205), and the National Natural Science Foundation of China (No. 91318301, 61321491, 61100037). This material is partially supported by the National Science Foundation under Grant No. IIS1017415, IIS-1162374, and IIS-1218036, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-11-C-0200 and W911NF-12-C-0028, and by Region II University Transportation Center under the project number 49997-33 25.

The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government

is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

9. REFERENCES

- [1] S. S. Anand and N. Griffiths. A market-based approach to address the new item problem. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 205–212, 2011.
- [2] P. Bogdanov, M. Busch, J. Moehlis, A. K. Singh, and B. K. Szymanski. Modeling individual topic-specific behavior and influence backbone networks in social media. *Social Network Analysis and Mining*, 4, 2014.
- [3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.
- [4] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *The Journal of Machine Learning Research*, 9:203–233, 2008.
- [5] H.-C. Chen and A. L. Chen. A music recommendation system based on music data grouping and user interests. In *CIKM*, pages 231–238, 2001.
- [6] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, pages 271–280, 2007.
- [7] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135, 2006.
- [8] Y. Fang and L. Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *HetRec*, pages 65–69, 2011.
- [9] Q. Gu, J. Zhou, and C. H. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210. SIAM, 2010.
- [10] S. Hacker and L. Von Ahn. Matchin: eliciting user preferences with an online game. In *CHI*, pages 1207–1216. ACM, 2009.
- [11] A. S. Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *SIGIR*, pages 91–98. ACM, 2008.
- [12] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.
- [13] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142. ACM, 2010.
- [14] M. Jiang, P. Cui, F. Wang, Q. Yang, W. Zhu, and S. Yang. Social recommendation across multiple relational domains. In *CIKM*, pages 1422–1431. ACM, 2012.
- [15] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434. ACM, 2008.
- [16] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [17] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.
- [18] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. In *CIKM*, pages 959–968, 2010.
- [19] H. Ma. An experimental study on implicit social recommendation. In *SIGIR*, pages 73–82. ACM, 2013.
- [20] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.
- [21] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296. ACM, 2011.
- [22] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.
- [23] R. Pan and M. Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *KDD*, pages 667–676, 2009.
- [24] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511, 2008.
- [25] U. Paquet and N. Koenigstein. One-class collaborative filtering with random graphs. In *WWW*, pages 999–1008, 2013.
- [26] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28, 2009.
- [27] A. Popescu-Belis and N. Pappas. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *36th ACM SIGIR Conference on Research and Development in Information Retrieval*, number EPFL-CONF-192567. ACM, 2013.
- [28] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
- [29] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260. ACM, 2002.
- [30] Y. Shen and R. Jin. Learning personal+ social latent factor model for social recommendation. In *KDD*, pages 1303–1311, 2012.
- [31] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic. One-class matrix completion with low-density factorizations. In *ICDM*, pages 1055–1060, 2010.
- [32] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha. Learning multiple-question decision trees for cold-start recommendation. In *WSDM*, pages 445–454. ACM, 2013.
- [33] J. Tang, H. Gao, and H. Liu. mTrust: discerning multi-faceted trust in a connected world. In *WSDM*, pages 93–102. ACM, 2012.
- [34] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *IJCAI*, pages 2712–2718, 2013.
- [35] J. Tang, H. Liu, H. Gao, and A. Das Sarma. etrust: understanding trust evolution in an online world. In *KDD*, pages 253–261. ACM, 2012.
- [36] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *WWW*, pages 537–546. ACM, 2011.
- [37] M. Zhang, J. Tang, X. Zhang, and X. Xue. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *SIGIR*, 2014.
- [38] X. Zheng, H. Ding, H. Mamitsuka, and S. Zhu. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *KDD*, pages 1025–1033, 2013.

[39] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *SIGIR*, pages 315–324, 2011.

APPENDIX

Proof of Theorem 2

By ignoring constant terms, we can re-write Eq. (7) as

$$J(\mathbf{F}) = -2\text{tr}[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G}\mathbf{F}'] + \text{tr}[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G}\mathbf{F}'] + \lambda_r \text{tr}(\mathbf{F}\mathbf{F}') + \lambda_F \text{tr}(\mathbf{F}'\mathbf{D}_M\mathbf{F}) - \lambda_F \text{tr}(\mathbf{F}'\mathbf{M}\mathbf{F}) \quad (15)$$

Following the auxiliary function approach [17], an auxiliary function $H(\mathbf{F}, \tilde{\mathbf{F}})$ of $J(\mathbf{F})$ must satisfy

$$H(\mathbf{F}, \mathbf{F}) = J(\mathbf{F}), \quad H(\mathbf{F}, \tilde{\mathbf{F}}) \geq J(\mathbf{F}) \quad (16)$$

We define

$$\mathbf{F}^{(t+1)} = \arg \min_{\mathbf{F}} H(\mathbf{F}, \mathbf{F}^{(t)}) \quad (17)$$

Then, by construction, we have

$$J(\mathbf{F}^{(t)}) = H(\mathbf{F}^{(t)}, \mathbf{F}^{(t)}) \geq H(\mathbf{F}^{(t+1)}, \mathbf{F}^{(t)}) \geq J(\mathbf{F}^{(t+1)}) \quad (18)$$

This would prove that $J(\mathbf{F}^{(t)})$ is monotonically decreasing.

In the remainder of proof, we need to find 1) an appropriate auxiliary function, and 2) the global minimum solution of the auxiliary function.

We start with the auxiliary function, and show that the following equation is one of the auxiliary functions for Eq. (15)

$$\begin{aligned} H(\mathbf{F}, \tilde{\mathbf{F}}) &= -2 \sum_{u=1}^m \sum_{k=1}^r [(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G}](u, k) \tilde{\mathbf{F}}(u, k) \\ &\quad (1 + \log(\frac{\mathbf{F}(u, k)}{\tilde{\mathbf{F}}(u, k)})) \\ &\quad - \sum_{u=1}^m \sum_{v=1}^m \sum_{k=1}^r \lambda_F \mathbf{M}(u, v) \tilde{\mathbf{F}}(v, k) \tilde{\mathbf{F}}(u, k) \\ &\quad (1 + \log(\frac{\mathbf{F}(v, k)\mathbf{F}(u, k)}{\tilde{\mathbf{F}}(v, k)\tilde{\mathbf{F}}(u, k)})) \\ &\quad + \sum_{u=1}^m \sum_{k=1}^r \lambda_r \mathbf{F}^2(u, k) \\ &\quad + \sum_{u=1}^m \sum_{k=1}^r \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\tilde{\mathbf{F}}\mathbf{G}'))\mathbf{G}](u, k) \mathbf{F}^2(u, k)}{\tilde{\mathbf{F}}(u, k)} \\ &\quad + \sum_{u=1}^m \sum_{k=1}^r \frac{[\lambda_F \mathbf{D}_M \tilde{\mathbf{F}}](u, k) \mathbf{F}^2(u, k)}{\tilde{\mathbf{F}}(u, k)} \end{aligned} \quad (19)$$

For convenience, we name the five terms in Eq. (19) as $E1$, $E2$, $E3$, $E4$ and $E5$, respectively. Then, for $E3$ we have

$$E3 = \lambda_r \text{tr}(\mathbf{F}\mathbf{F}') \quad (20)$$

Using the inequality $z \geq 1 + \log z$, we have

$$\begin{aligned} E1 &\geq -2 \sum_{u=1}^m \sum_{k=1}^r [(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G}](u, k) \mathbf{F}(u, k) \\ &= -2\text{tr}[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G}\mathbf{F}'] \end{aligned} \quad (21)$$

and

$$E2 \geq - \sum_{u=1}^m \sum_{v=1}^m \sum_{k=1}^r \lambda_F \mathbf{M}(u, v) \mathbf{F}(v, k) \mathbf{F}(u, k) = -\lambda_F \text{tr}(\mathbf{F}'\mathbf{M}\mathbf{F}) \quad (22)$$

For $E5$, we use the following inequality [7]

$$\sum_{i=1}^n \sum_{p=1}^k \frac{(\mathbf{A}\mathbf{S}^*\mathbf{B})\mathbf{S}^2(i, p)}{\mathbf{S}(i, p)} \geq \text{tr}(\mathbf{S}^*\mathbf{A}\mathbf{S}\mathbf{B})$$

where $\mathbf{A}_{n \times n}$, $\mathbf{B}_{k \times k}$, $\mathbf{S}_{n \times k}$, and $\mathbf{S}_{n \times k}^*$ are non-negative matrices, and \mathbf{A} and \mathbf{B} are symmetric. Therefore, we have

$$E5 \geq \lambda_F \text{tr}(\mathbf{F}'\mathbf{D}_M\mathbf{F}) \quad (23)$$

Finally, for $E4$, let $\mathbf{F}(u, k) = \tilde{\mathbf{F}}(u, k)\mathbf{Q}(u, k)$ we have

$$\begin{aligned} E4 &= \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^r \sum_{l=1}^r \frac{\tilde{\mathbf{F}}(u, l)\mathbf{G}'(l, i)\mathbf{W}^2(u, i)\mathbf{G}(i, k)\mathbf{F}^2(u, k)}{\tilde{\mathbf{F}}(u, k)} \\ &= \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^r \sum_{l=1}^r \tilde{\mathbf{F}}(u, l)\mathbf{G}'(l, i)\mathbf{W}^2(u, i)\mathbf{G}(i, k)\tilde{\mathbf{F}}(u, k)\mathbf{Q}^2(u, k) \\ &= \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^r \sum_{l=1}^r \tilde{\mathbf{F}}(u, l)\mathbf{G}'(l, i)\mathbf{W}^2(u, i)\mathbf{G}(i, k)\tilde{\mathbf{F}}(u, k) \\ &\quad (\frac{\mathbf{Q}^2(u, k) + \mathbf{Q}^2(u, l)}{2}) \\ &\geq \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^r \sum_{l=1}^r \tilde{\mathbf{F}}(u, l)\mathbf{G}'(l, i)\mathbf{W}^2(u, i)\mathbf{G}(i, k)\tilde{\mathbf{F}}(u, k) \\ &\quad (\mathbf{Q}(u, k)\mathbf{Q}(u, l)) \\ &= \sum_{u=1}^m \sum_{i=1}^n \sum_{k=1}^r \sum_{l=1}^r \mathbf{F}(u, l)\mathbf{G}'(l, i)\mathbf{W}^2(u, i)\mathbf{G}(i, k)\mathbf{F}(u, k) \\ &= \text{tr}[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{F}\mathbf{G}'))\mathbf{G}\mathbf{F}'] \end{aligned} \quad (24)$$

By substituting Eq. (20)-(24) into Eq. (19), we have $H(\mathbf{F}, \tilde{\mathbf{F}}) \geq J(\mathbf{F})$.

Next, we need to find the global minimum solution of $H(\mathbf{F}, \tilde{\mathbf{F}})$. The gradient is

$$\begin{aligned} \frac{1}{2} \frac{\partial H(\mathbf{F}, \tilde{\mathbf{F}})}{\partial \mathbf{F}(u, k)} &= - \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G}](u, k) \tilde{\mathbf{F}}(u, k)}{\mathbf{F}(u, k)} \\ &\quad - \frac{[\lambda_F \mathbf{M}\tilde{\mathbf{F}}](u, k) \tilde{\mathbf{F}}(u, k)}{\mathbf{F}(u, k)} + \frac{[\lambda_r \tilde{\mathbf{F}}](u, k) \mathbf{F}(u, k)}{\tilde{\mathbf{F}}(u, k)} \\ &\quad + \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\tilde{\mathbf{F}}\mathbf{G}'))\mathbf{G}](u, k) \mathbf{F}(u, k)}{\tilde{\mathbf{F}}(u, k)} \\ &\quad + \frac{[\lambda_F \mathbf{D}_M \tilde{\mathbf{F}}](u, k) \mathbf{F}(u, k)}{\tilde{\mathbf{F}}(u, k)} \\ &= - \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + \lambda_F \mathbf{M}\tilde{\mathbf{F}}](u, k) \tilde{\mathbf{F}}(u, k)}{\mathbf{F}(u, k)} \\ &\quad + \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\tilde{\mathbf{F}}\mathbf{G}'))\mathbf{G} + \lambda_r \tilde{\mathbf{F}} + \lambda_F \mathbf{D}_M \tilde{\mathbf{F}}](u, k) \mathbf{F}(u, k)}{\tilde{\mathbf{F}}(u, k)} \end{aligned} \quad (25)$$

We can further show that the Hessian matrix of $H(\mathbf{F}, \tilde{\mathbf{F}})$ is a diagonal matrix with positive diagonal elements. Therefore, the global minimum can be obtained by setting Eq. (25) as zero, which results in

$$\mathbf{F}^2(u, k) = \tilde{\mathbf{F}}^2(u, k) \frac{[(\mathbf{W} \odot \mathbf{W} \odot (\mathbf{R} + \mathbf{P}))\mathbf{G} + \lambda_F \mathbf{M}\tilde{\mathbf{F}}](u, k)}{[(\mathbf{W} \odot \mathbf{W} \odot (\tilde{\mathbf{F}}\mathbf{G}'))\mathbf{G} + \lambda_r \tilde{\mathbf{F}} + \lambda_F \mathbf{D}_M \tilde{\mathbf{F}}](u, k)} \quad (26)$$

Back to Eq. (17), $\mathbf{F}^{(t+1)} = \mathbf{F}$ and $\mathbf{F}^{(t)} = \tilde{\mathbf{F}}$. Therefore, the updating rule in Eq. (9) decreases monotonically. Further, with equivalence between Eq. (9) and Eq. (11) as shown in the proof of Theorem 1, we have that Eq. (7) decreases monotonically under the updating rule of Eq. (11), which completes the proof. \square