Toward Dual Roles of Users in Recommender Systems

Suhang Wang[†], Jiliang Tang[‡], and Huan Liu[†]
[†]Computer Science & Engineering, Arizona State University, Tempe, AZ, USA
[‡]Yahoo Labs, Sunnyvalue, CA, USA
[†]{suhang.wang,huan.liu}@asu.edu, [‡]jlt@yahoo-inc.com

ABSTRACT

Users usually play dual roles in real-world recommender systems. One is as a reviewer who writes reviews for items with rating scores, and the other is as a rater who rates the helpfulness scores of reviews. Traditional recommender systems mainly consider the reviewer role while not taking into account the rater role. However, the rater role allows users to express their opinions toward reviews about items; hence it may indirectly indicate their opinions about items, which could be complementary to the reviewer role. Since most real-world recommender systems provide convenient mechanisms for the rater role, recent studies show that typically there are much more *helpfulness ratings* from the rater role than *item ratings* from the reviewer role. Therefore, incorporating the rater role of users may have the potentials to mitigate the data sparsity and cold-start problems in traditional recommender systems. In this paper, we investigate how to exploit dual roles of users in recommender systems. In particular, we provide a principled way to exploit the rater role mathematically and propose a novel recommender system DualRec, which captures both the reviewer role and the rater role of users simultaneously for recommendation. Experimental results on two real world datasets demonstrate the effectiveness of the proposed framework, and further experiments are conducted to understand the importance of the rater role of users in recommendation.

Categories and Subject Descriptors

H.2.8 [**Database applications Subjects**]: Data mining; H.3.3 [**Information Search and Retrieval Subjects**]: Information filtering

General Terms

Algorithms

Keywords

Collaborative filtering; Helpfulness rating; Cold-start

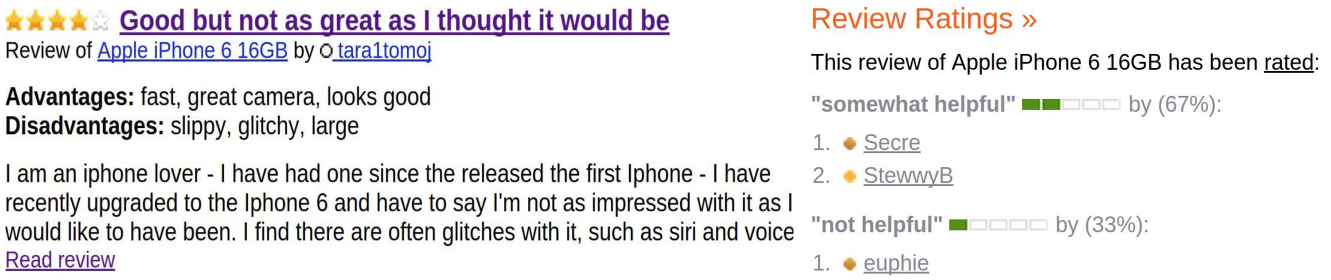
CIKM '2015 Melbourne, Australia
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806520.

1. INTRODUCTION

Recommender systems[22] intend to provide users with information of potential interest based on their demographic profiles and historical data. Collaborative Filtering (CF), which only requires past user ratings to predict unknown ratings, has attracted more and more attention[6, 33, 9]. Collaborative Filtering can be roughly categorized into memory-based[5, 23, 29] and model-based methods[6, 16, 10]. Memory-based methods mainly use the neighborhood information of users or items in the user-item rating matrix while model-based methods usually assume that an underlying model governs the way users rate items, and in general, it has better performance than memory-based methods. Despite the success of various model-based methods [24, 6], matrix factorization (MF) based model has became one of the most popular methods due to its good performance and efficiency in handling large datasets[25, 16, 10, 4, 27, 2].

Users in real-world recommender systems can play as reviewers that write reviews and give ratings for items, and they also can play as raters that rate the helpfulness of reviews. Figure 1 contains two snapshots from a real-world site Ciao[1]. In Figure 1(a)[2] , a user *tara1tomoj*, as a reviewer, writes a review for Apple iPhone 6 16GB, which gives a rating score 4 to iPhone 6. Figure 1(b)[3] shows that other users, as raters, rate the helpfulness of the review in Figure 1(a) - three raters give helpfulness ratings to this review, among which two give somewhat helpful (or score 2) and one gives not helpful (or score 1). However, the vast majority of traditional recommender systems only exploit the reviewer role while overlook the rater role of users. The rater role enables users to conveniently express their personal opinions on reviews about items, which indicates their indirect opinions on items. For example, though we don't know the exact rating scores these three users may give to Apple iPhone 6 16GB, low helpfulness ratings given by these users imply that they don't agree with the review; hence, they are likely to give some ratings different from 4 to Apple iPhone 6. Therefore, the rater role could be complementary to the reviewer role for recommendation. Since real-world recommender systems often provide convenient mechanisms for the rater role, such as clicking helpfulness buttons in Amazon and specifying helpfulness scores in Ciao, recent studies suggest that users usually have more helpfulness ratings produced by their rater role than item ratings

[1]www.ciao.com
[2]http://www.ciao.co.uk/Recent_Reviews/Top100/Smartph-ones_Mobile_Phones_5302356_2/All
[3]http://www.ciao.co.uk/Apple_iPhone_6__Review_6146860



(a) A Product Review Example (b) Helpfulness Ratings of the Review

Figure 1: An Example of User's Dual Roles in Recommender Systems.

produced by their reviewer role [26]. For example, for those users with few item ratings, they could have many helpfulness ratings. This property of the rater role of users could be useful to mitigate the data sparsity and cold-start problems, which are two major challenges of traditional recommender systems[18]. Therefore, incorporating the rater role of users has potentials to improve the performance of recommender systems.

In this paper, we investigate the dual roles of users, i.e. the reviewer role and the rater role, in recommender systems. In essence, we study two challenges - (1) how to capture the rater role of users mathematically; and (2) how to exploit the dual roles of users simultaneously for recommendation. In an attempt to solve these two challenges, we propose a novel recommendation framework DualRec. The major contributions of this paper are summarized next:

- Providing a principled way to capture the rater role of users mathematically;
- Proposing a novel recommender system DualRec, which exploits the duel roles of users into a coherent model for recommendation; and
- Conducting experiments on real-world datasets to understand the effectiveness of the proposed framework DualRec.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed framework DualRec with the details of how to capture the rater role of users and how to incorporate it with the reviewer role for recommendation. In Section 3, we present a method to solve the optimization problem of DualRec along with the time complexity analysis. In Section 4, we show empirical evaluation with discussion. In Section 5, we present the related work. In Section 6, we give conclusion with future work.

2. A RECOMMENDER SYSTEM WITH DUAL ROLES OF USERS

Before introducing details about the proposed framework, we first introduce notations used in this paper. Throughout this paper, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, $\mathbf{M}_{ij}$ denotes the (i,j)-th entry of $\mathbf{M}$ while $\mathbf{m}_i$ and $\mathbf{m}^j$ mean the i-th row and j-th column of $\mathbf{M}$, respectively. $||\mathbf{M}||_F$ is the Frobenius norm of $\mathbf{M}$. Capital letters in calligraphic math font such as $\mathcal{P}$ are used to denote sets. We use $|\cdot|$ to denote the cardinality of a set, for example, $|\mathcal{P}|$ indicates the number of elements in the set $\mathcal{P}$.

Typically there are three types of objects, namely, users, reviews and items. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users, $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ be the set of items and $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ be the set of reviews where n, m and N are the numbers of users, items and reviews, respectively. As reviewers, users can write reviews and rate items. For example, (u_i, r_j, p_k) means user u_i writes a review r_j which gives a rating to item p_k. We use the matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ to denote the user-item rating matrix produced by the reviewer role of users, where $\mathbf{R}_{ik}$ is the rating score if u_i rates p_k otherwise $\mathbf{R}_{ik}$ = "?" for an unknown rating from u_i to p_k. $\mathbf{A} \in \{0,1\}^{N\times m}$ is the review item matrix where $\mathbf{A}_{jk} = 1$ if review r_j is associated with item p_k and $\mathbf{A}_{jk} = 0$ otherwise. As raters, users can rate the helpfulness of reviews and $\mathbf{H} \in \mathbb{R}^{n\times N}$ is employed to represent user review helpfulness rating matrix where $\mathbf{H}_{ij}$ is the helpfulness score if user u_i gives a review r_j a rating and $\mathbf{H}_{ij}$ = "?" denotes an unknown helpfulness score from u_i to r_j. We use $\mathcal{O}(\mathbf{R}) = \{\mathbf{R}_{ij} | \mathbf{R}_{ij} = ?\}$ to denote the set of unknown item ratings. Next we begin the introduction of the proposed framework with the basic model to exploit the reviewer role of users.

2.1 A Basic Model for the Reviewer Role of Users

We choose matrix factorization based collaborative filtering as the basic model to exploit the reviewer role of users since it is a very practical and popular method to build recommender systems [6, 25, 16, 7, 10, 27]. It tries to map both users and items to a joint latent factor space with dimensionality K such that user-item interactions are modeled as inner products in that space[10]. The premise behind a low-dimensional factor model is that there is only a small number of factors that influence the user preferences and a user's preference vector is determined by how each factor applies to the user[25]. Specifically, given the rating matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$, matrix factorization methods try to find two matrices $\mathbf{U} \in \mathbb{R}^{K\times n}$ and $\mathbf{V} \in \mathbb{R}^{K\times m}$ by solving the following optimization problem

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{I}_{ij}(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})^2 + \gamma(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \quad (1)$$

where $\mathbf{I} \in \{0,1\}^{n\times m}$ is the indicator matrix defined as

$$\mathbf{I}_{ij} = \begin{cases} 1 & \text{if } \mathbf{R}_{ij} \neq \text{"?"} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

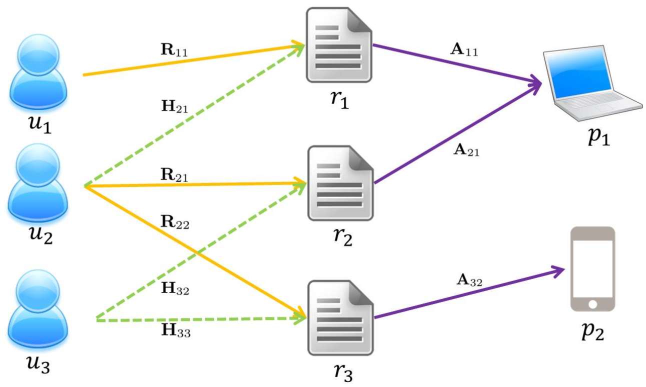


Figure 2: The figure gives an example of user's dual roles in recommender systems. The yellow arrow means that a user, as a reviewer, writes a review and the review assigns a rating score to a product connected by a purple arrow. A green (dashed) arrow denotes that a user, as a rater, gives a helpfulness rating score to a review.

$\mathbf{U}$ is the user latent factor matrix with each column $\mathbf{u}_i$ being u_i's latent factor and $\mathbf{V}$ is the item latent factor matrix with each column $\mathbf{v}_j$ being the latent property of the item p_j. The term $\gamma(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$ is introduced to avoid overfitting. The rating of u_i to p_j is predicted as $\mathbf{u}_i^T\mathbf{v}_j$. With the basic model for the reviewer role, next we will introduce our solution to capture the rater role and the proposed framework DualRec.

2.2 Capturing the Rater Role of Users

Helpfulness ratings produced by the rater role of users indicate users' opinions toward reviews about items and it could reflect their opinions on items indirectly. Therefore, the key problem to capture the rater role of users for recommendation is how to learn implicit item ratings from helpfulness ratings produced by the rater role. Next we first introduce some definitions and then detail the mathematical model to capture the rater role of users.

Assume that a review r_k is about the item p_j, i.e, $\mathbf{A}_{kj} = 1$. A rater u_i who rates r_k about p_j may or may not be a reviewer of p_j. For example, as shown in Figure 2, user u_2 rates the helpfulness of a review r_1 about item p_1, and he/she also directly writes a review r_2 and gives an item rating to p_1; while u_3 only rates the helpfulness of a review r_2 about p_1. Therefore, for a helpfulness rating from u_i to r_k about p_j, i.e. $\mathbf{H}_{ik}$, according to whether u_i writes reviews about p_j or not, we can assign it as either *Type-I* helpfulness rating or *Type-II* helpfulness rating . The definitions of these two types of helpfulness ratings are given as follows:

Definition 1. *A helpfulness rating $\mathbf{H}_{ik}$ is a Type-I helpfulness rating if the item p_j associated with the review r_k in $\mathbf{H}_{ik}$ is also rated by user u_i. In other words, u_i not only rates the helpfulness of the review r_k about p_j but also directly gives a rating score to p_j . Formally, a Type-I helpfulness rating $\mathbf{H}_{ik}$ satisfies:*

$$\exists p_j \in \mathcal{P}, \quad \mathbf{H}_{ik} > 0 \ \wedge \ \mathbf{A}_{kj} = 1 \ \wedge \ \mathbf{R}_{ij} \neq ? \tag{3}$$

In Figure 2, $\mathbf{H}_{21}$ is a Type-I helpfulness rating since u_2 wrote r_2 which assigns a rating to p_1.

Definition 2. *A helpfulness rating $\mathbf{H}_{ik}$ is a Type-II helpfulness rating if the item p_j associated with the review r_k in $\mathbf{H}_{ik}$ is not rated by user u_i. In other words, u_i only rates the helpfulness of the review r_k about p_j but not the item p_j. Formally, a Type-II helpfulness rating $\mathbf{H}_{ik}$ satisfies:*

$$\nexists p_j \in \mathcal{P}, \quad \mathbf{H}_{ik} > 0 \ \wedge \ \mathbf{A}_{kj} = 1 \ \wedge \ \mathbf{R}_{ij} \neq ? \tag{4}$$

In Figure 2, $\mathbf{H}_{32}$ and $\mathbf{H}_{33}$ are Type-II helpfulness ratings because u_3 doesn't rate any items.

We further use $\mathcal{P}_{ij}$ to represent the set of reviews about p_j that received Type-I helpfulness ratings from u_i. That is, $\forall r_k \in \mathcal{P}_{ij}$, we have that r_k is about p_j, and u_i rates both r_k and also p_j. Formally, $\mathcal{P}_{ij}$ is defined as:

$$\begin{aligned} \mathcal{P}_{ij} = \{r_k | &\mathbf{H}_{ik} \text{ is a Type-I helpfulness rating} \\ &\wedge \mathbf{A}_{kj} = 1, 1 \le k \le N\} \end{aligned} \tag{5}$$

For example, in Figure 2, $\mathcal{P}_{21} = \{r_1\}$ because u_2 rated both p_1 and r_1 of p_1, while $\mathcal{P}_{31} = \emptyset$ because u_3 doesn't give a rating to p_1.

Similarly, we use $\mathcal{Q}_{ij}$ to represent a set of reviews about p_j which received Type-II helpfulness ratings from u_i. That is, $\forall r_k \in \mathcal{P}_{ij}$, we have that r_k is about p_j, and u_i rates r_k but not p_j. Formally, $\mathcal{Q}_{ij}$ is defined as:

$$\begin{aligned} \mathcal{Q}_{ij} = \{r_k | &\mathbf{H}_{ik} \text{ is a Type-II helpfulness rating} \\ &\wedge \mathbf{A}_{kj} = 1, 1 \le k \le N\}\} \end{aligned} \tag{6}$$

For example, in Figure 2, $\mathcal{Q}_{21} = \emptyset$ because while $\mathcal{Q}_{31} = \{r_2\}$.

For a review r_k that is written by u_s and is associated with the item p_j, $\mathbf{R}_{sj}$ is the item rating from u_s to p_j. The helpfulness rating $\mathbf{H}_{ik}$ indicates the opinion of user u_i on the review r_k and the corresponding item rating $\mathbf{R}_{sj}$ from u_s to p_j. Therefore, both the helpfulness rating $\mathbf{H}_{ik}$ and the item rating from the author of r_k to p_j, i.e., $\mathbf{R}_{sj}$, could be useful to learn the implicit item rating from u_i to p_j. Let $\phi(\mathbf{H}_{ik}, \mathbf{R}_{sj}) \in \mathbb{R}^{d\times 1}$ be the feature vector for a helpfulness rating from u_i to r_k[4]. We further assume a linear map matrix $\mathbf{w}$ which predicts the implicit item rating $\hat{\mathbf{R}_{ij}}$ from $\phi(\mathbf{H}_{ik}, \mathbf{R}_{sj})$ as $\hat{\mathbf{R}_{ij}} = \phi(\mathbf{H}_{ik}, \mathbf{R}_{sj})^T\mathbf{w}$. Therefore, we propose the following minimization terms to capture the rater role of users for recommendation as:

$$\begin{aligned} \min_{\mathbf{w}} \quad &\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{F}_{ij}(\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}} \phi(\mathbf{H}_{ik}, \mathbf{R}_{sj})^T\mathbf{w} - \mathbf{R}_{ij})^2 \\ + \beta &\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{G}_{ij}(\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}} \phi(\mathbf{H}_{ik}, \mathbf{R}_{sj})^T\mathbf{w} - \mathbf{u}_i^T\mathbf{v}_j)^2 \end{aligned} \tag{7}$$

Next we give details about the inner working of Eq.(7) as:

- For a user-item pair (u_i, p_j) with $\mathcal{P}_{ij} \neq \emptyset$, u_i not only rates the helpfulness of reviews about p_j but also gives a rating score to p_j; hence the first term in Eq.(7) ensures the consistency between the predicted item ratings from the rater role (or $\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}} \phi(\mathbf{H}_{ik}, \mathbf{R}_{sj})^T\mathbf{w}$) and the observed item ratings from the reviewer role (or $\mathbf{R}_{ij}$). Since a user u_i may rate the helpfulness of

[4]The d-dimensional feature vector could encode features from users, reviews and pairs of item and helpfulness ratings. In this paper, we empirically find that some simple features from a pair of item and helpfulness ratings, i.e., $\phi(x, y) = [x; y; xy; x^2; y^2; 1/x; 1/y]$, works well. We would like to investigate other features in the future.

multiple reviews about the same item p_j, we use the average rating $\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}$ as the final predicted item rating from u_i to p_j. $\mathbf{F}$ is an indicator matrix, which is formally defined as

$$\mathbf{F}_{ij}=\begin{cases}1 & \text{if } |\mathcal{P}_{ij}|>0\\ 0 & \text{if } |\mathcal{P}_{ij}|=0\end{cases} \tag{8}$$

- For a user-item pair (u_i, p_j) with $\mathcal{Q}_{ij}\neq\emptyset$, u_i only rates the helpfulness of reviews about p_j and has no rating score to p_j; hence the second term in Eq.(7) ensures the consistency between the predicted item ratings from the rater role (or $\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}$) and the predicted item ratings from the reviewer role (or $\mathbf{u}_i^T\mathbf{v}_j$). Similar to $\mathbf{F}$, $\mathbf{G}$ is an indicator matrix defined as

$$\mathbf{G}_{ij}=\begin{cases}1 & \text{if } |\mathcal{Q}_{ij}|>0\\ 0 & \text{if } |\mathcal{Q}_{ij}|=0\end{cases} \tag{9}$$

2.3 The Proposed Framework–DualRec

With model components to capture the reviewer role and the rater role, the proposed recommender system DualRec that exploits the dual roles of users simultaneously is to solve the following optimization problem as:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{w}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{I}_{ij}(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})^2\\ & +\alpha\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}(\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}-\mathbf{R}_{ij})^2\\ & +\beta\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}(\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}-\mathbf{u}_i^T\mathbf{v}_j)^2\\ & +\gamma(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)+\lambda\|\mathbf{w}\|_2^2\end{aligned} \tag{10}$$

where the first term models the reviewer role of users based on matrix factorization; and the second and third terms incorporate the rater role of users. α and β are introduced to leverage the contribution of reviewer role and rater role. Similar to $\gamma(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)$, the term $\lambda\|\mathbf{w}\|_2^2$ is introduced to avoid overfitting.

If we only consider the reviewer role of users, we cannot learn the user latent factor $\mathbf{u}_i$ for a user u_i without any item rating scores. However, as shown in Eq.(10), we still can learn the user latent factor for u_i if u_i has helpfulness ratings by incorporating the rater role of users. Also both user latent factor matrix $\mathbf{U}$ and the item latent factor matrix $\mathbf{V}$ are learned from the dual roles of users. Therefore, the proposed framework DualRec has potentials to mitigate the cold-start and data sparsity problems in recommendation.

3. AN OPTIMIZATION METHOD FOR DUALREC

The objective function in Eq.(10) is not convex if we update all the variables jointly. To optimize the objective function, we use alternating least squares, which is a popular method for MF based collaborative filtering. Specifically, we optimize one variable by fixing other variables. Next, we give the details to optimize the objective function followed by the complexity analysis of the proposed algorithm.

3.1 Update Rule of U and V

To get the gradients of Eq.(10) w.r.t to $\mathbf{U}$ and $\mathbf{V}$, we first remove terms that are irrelevant to $\mathbf{U}$ and $\mathbf{V}$ and rewrite the objective function as

$$\begin{aligned}\mathcal{L}(\mathbf{U},\mathbf{V})=&\|\mathbf{I}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{R})\|_F^2+\gamma(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)\\ &+\beta\|\mathbf{G}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{M})\|_F^2\end{aligned} \tag{11}$$

where $\odot$ denotes Hadmard product and $\mathbf{M}$ is defined as

$$\mathbf{M}_{ij}=\begin{cases}\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}, & \text{if } \mathbf{G}_{ij}>0\\ 0, & \text{o.w.}\end{cases} \tag{12}$$

Then the gradient of $\mathcal{L}(\mathbf{U},\mathbf{V})$ with respect to $\mathbf{U}$ is given as

$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{V})}{\partial\mathbf{U}}=&2\mathbf{V}[\mathbf{I}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{R})]^T+2\gamma\mathbf{U}\\ &+2\beta\mathbf{V}[\mathbf{G}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{M})]^T\end{aligned} \tag{13}$$

Therefore, $\mathbf{U}$ is updated as

$$\mathbf{U}=\mathbf{U}-\epsilon\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{V})}{\partial\mathbf{U}} \tag{14}$$

where ϵ is the learning rate. Similarly, the gradient of $\mathcal{L}(\mathbf{U},\mathbf{V})$ with respect to $\mathbf{V}$ is given as

$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{V})}{\partial\mathbf{V}}=&2\mathbf{U}[\mathbf{I}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{R})]+2\gamma\mathbf{V}\\ &+2\beta\mathbf{U}[\mathbf{G}\odot(\mathbf{U}^T\mathbf{V}-\mathbf{M})]\end{aligned} \tag{15}$$

and $\mathbf{V}$ is updated as

$$\mathbf{V}=\mathbf{V}-\epsilon\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{V})}{\partial\mathbf{V}} \tag{16}$$

3.2 Update Rule of w

Similarly, to get the gradient of Eq.(10) w.r.t to $\mathbf{w}$, we first remove terms that are irrelevant to $\mathbf{w}$ and the objective is simplified as

$$\begin{aligned}\mathcal{L}(\mathbf{w})=&\alpha\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}(\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})^T\mathbf{w}-\mathbf{R}_{ij})^2\\ &+\beta\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}(\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{kj})^T\mathbf{w}-\mathbf{u}_i^T\mathbf{v}_j)^2\\ &+\lambda\|\mathbf{w}\|_2^2\end{aligned} \tag{17}$$

For simplicity, let $\mathbf{x}_{ij}=\frac{1}{|\mathcal{P}_{ij}|}\sum_{k\in\mathcal{P}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})$ and $\mathbf{z}_{ij}=\frac{1}{|\mathcal{Q}_{ij}|}\sum_{k\in\mathcal{Q}_{ij}}\phi(\mathbf{H}_{ik},\mathbf{R}_{sj})$. Then we have

$$\begin{aligned}\mathcal{L}(\mathbf{w})=&\alpha\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}(\mathbf{x}_{ij}^T\mathbf{w}-\mathbf{R}_{ij})^2+\lambda\|\mathbf{w}\|_2^2\\ &+\beta\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}(\mathbf{z}_{ij}^T\mathbf{w}-\mathbf{u}_i^T\mathbf{v}_j)^2\end{aligned} \tag{18}$$

The derivative of Eq.(18) with respect to $\mathbf{w}$ is given as

$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{w})}{\partial\mathbf{w}}=&2\alpha\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}\mathbf{x}_{ij}(\mathbf{x}_{ij}^T\mathbf{w}-\mathbf{R}_{ij})\\ &+2\beta\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}\mathbf{z}_{ij}(\mathbf{z}_{ij}^T\mathbf{w}-\mathbf{u}_i^T\mathbf{v}_j)+2\lambda\mathbf{w}\end{aligned} \tag{19}$$

By setting the derivative to zero, we have the update rule for $\mathbf{w}$ as

$$\mathbf{w} =(\alpha \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{F}_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^T + \beta \sum_{i}\sum_{j} \mathbf{G}_{ij}\mathbf{z}_{ij}\mathbf{z}_{ij}^T + \lambda \mathbf{I})^{-1} (\alpha \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{F}_{ij}\mathbf{R}_{ij}\mathbf{x}_{ij} + \beta \sum_{i}\sum_{j} \mathbf{G}_{ij}\mathbf{u}_i^T\mathbf{v}_j\mathbf{z}_{ij}) \tag{20}$$

Algorithm 1 DualRec

Input: $\mathbf{R} \in \mathbb{R}^{n\times m}, \mathbf{H} \in \mathbb{R}^{n\times N}, \mathbf{A} \in \{0,1\}^{N\times m}, K, \alpha, \beta, \gamma$
Output: $\hat{\mathbf{R}}$

1: **for** i=1 to n **do**
2:    **for** j=1 to m **do**
3:       Construct $\mathcal{P}_{ij}$ according to Eq.(5)
4:       Construct $\mathcal{Q}_{ij}$ according to Eq.(6)
5:    **end for**
6: **end for**
7: Construct indicator matrix $\mathbf{I}, \mathbf{F}, \mathbf{G}$
8: Initialize $\mathbf{U} \in \mathbb{R}^{K\times n}, \mathbf{V} \in \mathbb{R}^{K\times m}$
9: Initialize $\mathbf{w}$ by solving Eq.(7) with β set to 0
10: **repeat**
11:    Calculate $\mathbf{M}$ using Eq.(12)
12:    Calculate $\frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{U}}$ using Eq.(13)
13:    Update $\mathbf{U}$ as $\mathbf{U} = \mathbf{U} - \epsilon \frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{U}}$
14:    Calculate $\frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{V}}$ using Eq.(15)
15:    Update $\mathbf{V}$ as $\mathbf{V} = \mathbf{V} - \epsilon \frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{V}}$
16:    Update $\mathbf{w}$ using Eq.(20)
17: **until** Convergence
18: Calculate $\hat{\mathbf{R}} = \mathbf{U}^T\mathbf{V}$
19: return $\hat{\mathbf{R}}$

3.3 The Algorithm

With the update rules of $\mathbf{U}, \mathbf{V}$ and $\mathbf{w}$ given above, the optimization algorithm for DualRec is shown in Algorithm 1. Next we briefly review Algorithm 1. For each user-item pair, we first find $\mathbf{P}_{ij}$ and $\mathbf{Q}_{ij}$ from line 1 to line 6. Based on $\mathbf{R}, \mathbf{P}_{ij}$ and $\mathbf{Q}_{ij}$, we can construct the indicator matrix $\mathbf{I}, \mathbf{F}$ and $\mathbf{G}$ in line 7. We randomly initialize $\mathbf{U}$ and $\mathbf{V}$ in line 8. In order to speed up the learning process, $\mathbf{w}$ is initialized by solving Eq.(7) with $\beta = 0$ since it can give a better approximation to $\mathbf{w}$ than randomly guessing in line 9. After initialization, $\mathbf{U}, \mathbf{V}$ and $\mathbf{w}$ are updated sequentially until it converges from line 10 to line 17. Finally, the rating matrix is reconstructed as $\hat{\mathbf{R}} = \mathbf{U}^T\mathbf{V}$. With the reconstructed matrix $\hat{\mathbf{R}}$, an unknown rating from u_i to p_j is predicted as $\hat{\mathbf{R}}_{ij}$

3.4 Complexity Analysis

The algorithm is composed of two parts, i.e., initialization and updating. The most time consuming part is the updating part. The computational cost of $\mathbf{U}^T\mathbf{V}$ in Eq.(13) is $\mathcal{O}(nkm)$. Considering the fact that $\mathbf{I}$ is very sparse, we have that $\mathbf{I} \odot (\mathbf{U}^T\mathbf{V} - \mathbf{R})$ is also very sparse. Thus, the computational cost of $\mathbf{V}[\mathbf{I} \odot (\mathbf{U}^T\mathbf{V} - \mathbf{R})]^T$ is about $\mathcal{O}(nkm)$. Then the computational cost of $\frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{U}}$ using Eq.(15) is $\mathcal{O}(nkm)$. Similarly, the computational cost of $\frac{\partial \mathcal{L}(\mathbf{U},\mathbf{V})}{\partial \mathbf{V}}$ using Eq.(15) is also $\mathcal{O}(nmk)$. To update $\mathbf{w}$ using Eq.(20), we can pre-calculate $(\alpha \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{F}_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^T + \beta \sum_i\sum_j \mathbf{G}_{ij}\mathbf{z}_{ij}\mathbf{z}_{ij}^T + \lambda \mathbf{I})^{-1}$ and $\alpha \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{F}_{ij}\mathbf{R}_{ij}\mathbf{x}_{ij}$ since they are fixed in each iteration. Also, considering the fact that $\mathbf{G}$ is very sparse, the computational cost of $\beta \sum_i\sum_j \mathbf{G}_{ij}\mathbf{u}_i^T\mathbf{v}_j\mathbf{z}_{ij}$ is $\mathcal{O}(ks)$, where s is the number of nonzero elements in $\mathbf{G}$. Thus, the computational cost of $\mathbf{w}$ is $\mathcal{O}(ks)$. Thus the total cost of the algorithm is $\mathcal{O}(t(nkm + ks))$, where t is the number of iterations. From the analysis, we can see that the computational cost of Algorithm 1 in each iteration is actually comparable to that of the matrix factorization based collaborative filtering.

Table 1: Statistics of the Datasets

Dataset	Epinions	Ciao
# of users	2,161	2,368
# of items	2,796	3,046
# of ratings	72,971	69,453
# of H.R.	581,880	1,248,020
# of Type-I H.R.	47,905	177,038
# of Type-II H.R.	533,975	1,070,982

4. EXPERIMENTAL ANALYSIS

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework DualRec. Specifically, we aim to answer the following two questions:

- Can the proposed framework improve the recommendation performance by incorporating the rater role of users? and
- Is the proposed framework able to mitigate the cold-start problem for recommendation by incorporating the rater role of users?

We begin by introducing datasets and experimental settings, then we compare DualRec with the state-of-the-art recommendation systems to answer the first question and we investigate the capability of the proposed framework in handling the cold-start problem to answer the second question. Finally further experiments are conducted to investigate the sensitivity of DualRec to the parameters.

4.1 Datasets and Experimental Settings

We collect two datasets from real-world social media websites, i.e., Epinions[5] and Ciao. From the originally collected datasets, we filter out users who rated few items and also items that received less than 10 ratings. For both datasets, users can rate products and reviews with scores from 1 to 5. The statistics of the resulting datasets are shown in Table 1. In the table, H.R. denotes helpfulness rating, Type-I H.R. denotes Type-I helpfulness ratings and Type-II H.R indicates Type-II helpfulness ratings. It is evident from the statistics in the table that, on average, users have more helpfulness ratings from the rater role than item ratings from the reviewer role of users.

Two widely used evaluation metrics, i.e., mean absolute error (MAE) and root mean square error (RMSE), are adopted to evaluate the rating prediction performance. Specifically, MAE is defined as

$$MAE = \frac{\sum_{(i,j)\in\mathcal{T}} |\mathbf{R}_{ij} - \tilde{\mathbf{R}}_{ij}|}{|\mathcal{T}|} \tag{21}$$

[5]http://www.epinions.com/

Table 2: Performance comparison on Epinions and Ciao in terms of RMSE.

Dataset	Training Size	UCF	WNMF	MF	QMF	DualRec
Epinions	10%	1.3320	1.3581	1.3072	1.3068	1.2338
	20%	1.2642	1.2644	1.2196	1.2137	1.1784
	40%	1.2015	1.1812	1.14 01	1.1415	1.1201
Ciao	10%	1.2644	1.3128	1.2655	1.2551	1.1382
	20%	1.2233	1.2202	1.1652	1.1718	1.0920
	40%	1.1748	1.1378	1.0811	1.0828	1.0497

Table 3: Performance comparison on Epinions and Ciao in terms of MAE.

Dataset	Training Size	UCF	WNMF	MF	QMF	DualRec
Epinions	10%	1.0836	1.0756	1.0514	1.0575	1.0064
	20%	1.0086	0.9927	0.9678	0.9634	0.9403
	40%	1.0104	0.9248	0.8928	0.8901	0.8790
Ciao	10%	0.9385	1.0325	1.0087	1.0015	0.9049
	20%	0.9211	0.9522	0.9149	0.9278	0.8585
	40%	0.8873	0.8914	0.8361	0.8364	0.8187

and RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\mathcal{T}}\left(\mathbf{R}_{ij} - \tilde{\mathbf{R}}_{ij}\right)^2}{|\mathcal{T}|}} \tag{22}$$

where in both metrics, $\mathcal{T}$ denotes the set of ratings we want to predict, $\mathbf{R}_{ij}$ denotes the rating user u_i gives to item p_j and $\tilde{R}_{ij}$ denotes the predicted rating from $\mathbf{u}_i$ and $\mathbf{v}_j$.

For each dataset, we random select $x\%$ as the training set and the remaining $1 - x\%$ as the testing set. To investigate the capability of the proposed framework in handling the data sparsity problem, we vary x as $\{10, 20, 40\}$ in this work. The random selection process is carried out 10 times independently, and the average MAE and RMSE are reported. Note that previous work demonstrated that *small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendation*[8].

4.2 Performance Comparison of Recommender Systems

To answer the first question, we compare the proposed system with several representative systems. The comparison results are summarized in Tables 2 and 3 for RMSE and MAE, respectively. The representative systems in the table are defined as:

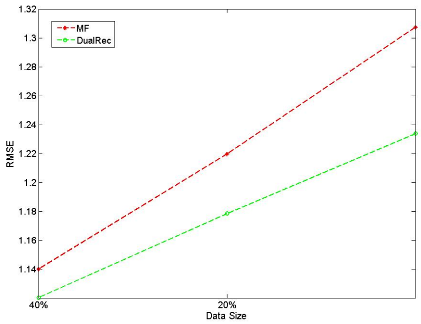
- **UCF**: UCF is the user-oriented collaborative filtering where the rating from u_i to p_j is predicted as an aggregation of ratings of K most similar users of u_i to p_j. We use the cosine similarity to calculate user-user similarity.
- **MF**: matrix factorization based collaborative filtering tries to decompose the user-item rating matrix into two matrices such that the reconstruction error is minimized [10]. In this work, we choose **MF** as the basic model of the proposed framework DualRec.
- **WNMF**: weighted nonnegative matrix factorization tries to decompose the weighted rating matrix into two nonnegative matrices to minimize the reconstruction error [33].
- **QMF**: Review quality aware collaborative filtering [21] uses helpfulness rating as a measure to capture the quality of the rating and then the quality scores $\mathbf{W}$ are incorporated as weights into matrix factorization based CF as

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{I}_{ij}(\mathbf{W}_{ij}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}))^2 + \gamma(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \tag{23}$$

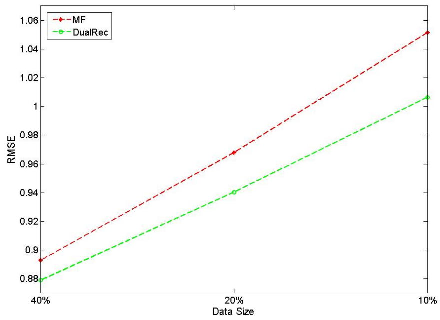
Note that parameters of all baseline methods are determined via cross validation. For DualRec, we set $\alpha = 1$, $\beta = 0.1$, $\gamma = 0.1$ and $\lambda = 1$ throughout the experiments. More details about parameter selection for the proposed framework will be discussed in the following subsections. From results in the Table 2 and 3, we make the following observations:

- In general, matrix factorization based recommender systems outperform the user-oriented CF method and this observation is consistent with that in [10].
- The proposed framework DualRec obtains better performance than baseline methods based on matrix factorization. We perform t-test on these results, which suggests that the improvement is significant. These results indicate that incorporating the rater role of users can improve the recommendation performance.
- The performance gaps between DualRec and MF with respect to the size of the training set are shown in Figure 3. We observe that the improvement of the proposed framework DualRec compared to MF increases as the datasets become sparser, i.e., from 40% to 10%. For example, the relative RMSE improvement of DualRec over MF is 2.90% on Ciao when the training size is 40% and increases to 10.06% when the training size is 10%. Compared to MF, the proposed framework is more robust to the data sparsity problem. These observations suggest that exploiting the rater role of users can mitigate the data sparsity problem for recommendation.

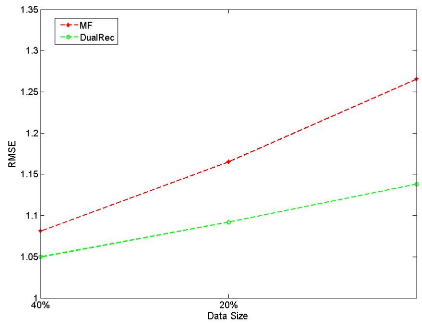
Via aforementioned analysis, we can draw an answer to the first question - incorporating the rater role of users not



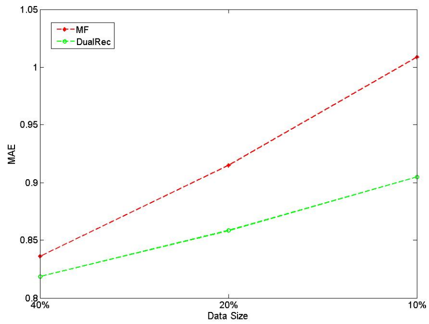
(a) RMSE on Epinions



(b) MAE on Epinions



(c) RMSE on Ciao



(d) MAE on Ciao

Figure 3: The Performance Gaps Between MF and the Proposed Framework DualRec w.r.t. the Size of the Training Set.

only can significantly improve the recommendation performance but also can mitigate the data sparsity problem in recommender systems.

4.3 Capability of Handling Cold-start Users

To answer the second question, we investigate the capability of the proposed framework DualRec in handling cold-start users. In detail, we first randomly select $x\%$ as training set and the remaining $1-x\%$ as testing set where x is also varied as $\{10, 20, 40\}$. Then, we randomly select 5% users from the training set and remove their item ratings from the training set to the test set. While the helpfulness ratings for these 5% users are kept. In this way, these 5% of users do not have any item ratings but may have helpfulness ratings and we consider these 5% of users as cold-start users. For those baseline methods that cannot handle cold-start users, we randomly guess their item ratings for cold-start users. The results on those training sets with cold-start users are summarized in Table 4 and Table 5 for RMSE and MAE, respectively. Note that numbers inside parentheses in Table 4 and Table 5 denote the performance reductions compared to the performance without cold-start users in Table 2 and Table 3, respectively.

From the tables, we make the following observations

- The performance of all methods degenerates when we introduce cold-start users. For example, the performance for MF decreases up to 5.75% in terms of RMSE and up to 6.35% in terms of MAE.
- Compared to the baseline methods, the performance degeneration of the proposed framework DualRec is much smaller. As aforementioned, the proposed framework can still learn user latent factors for cold-start users by exploiting the rater role of users. These results support that the proposed framework can mitigate cold-start problems for recommendation.

In summary, the introduction of cold-start users could degrade the recommendation performance and the proposed framework is relatively more robust to cold-start users by incorporating the rater role of users.

4.4 Parameter Sensitivity

The proposed framework has two importance parameters α and β, which together control the contributions of incorporating the rater role of users for recommendation. In this section, we investigate the impact of the parameters α and β on the performance of the proposed framework DualRec. We only show results on Epinions and Ciao with 20% and 40% without cold-start users since we have similar observations with other experimental settings. We empirically set the latent dimension $K = 10$, the regularization parameter $\gamma = 0.1$ and $\lambda = 1$. We vary the values of α as $\{0.001, 0.01, 0.1, 1, 10, 100\}$ and β as $\{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$. The results are shown in Figure4.

It can be observed from the figure:

- When we set $\alpha = 0$ and $\beta = 0$, the proposed framework DualRec boils down to MF. When we increase their values to $\alpha = 0.001$ and $\beta = 0.001$, we incorporate

Table 4: **Performance comparison on Epinions and Ciao with 5% cold-start users in terms of RMSE. Note that numbers inside parentheses in the table denote the performance reductions compared to the performance without cold-start users in Table 2.**

Dataset	Training Size	UCF	WNMF	MF	QMF	DualRec
Epinions	10%	1.3915(-4.47)	1.5899(-17.07)	1.3394(-2.46)	1.3296(-1.97)	1.2545(-1.68)
	20%	1.3475(-6.59)	1.5415(-21.91)	1.2459(-2.16)	1.2387(-2.06)	1.1927(-1.01)
	40%	1.3070(-8.78)	1.5347(-29.93)	1.1813(-3.62)	1.1746(-3.07)	1.1388(-1.62)
Ciao	10%	1.3035(-3.09)	1.6109(-22.70)	1.311(-3.60)	1.2879(-3.41)	1.1575(-1.66)
	20%	1.2713(-3.92)	1.5519(-27.18)	1.2125(-4.06)	1.1956(-2.88)	1.1044(-1.14)
	40%	1.2518(-6.55)	1.5404(-35.38)	1.1433(-5.75)	1.1338(-4.71)	1.0629(-1.24)

Table 5: **Performance comparison on Epinion and Ciao with 5% cold-start users in terms of MAE. Note that numbers inside parentheses in the table denote the performance reductions compared to the performance without cold-start users in Table 3.**

Dataset	Training Size	UCF	WNMF	MF	QMF	DualRec
Epinions	10%	1.1237(-3.70)	1.2091(-12.41)	1.0837(-3.07)	1.0754(-2.07)	1.0126(-0.30)
	20%	1.0961(-8.68)	1.1484(-15.68)	0.9917(-2.47)	0.9859(-2.34)	0.9528(-1.21)
	40%	1.0569(-4.60)	1.1123(-20.28)	0.9287(-4.02)	0.9203(-3.84)	0.9043(-2.85)
Ciao	10%	0.9747(-3.86)	1.1967(-15.90)	1.0494(-4.034)	1.0315(-3.69)	0.9238(-1.25)
	20%	0.9658(-4.85)	1.1286(-18.53)	0.9556(-4.45)	0.9383(-2.43)	0.8692(-1.25)
	40%	0.9582(-7.99)	1.1256(-26.27)	0.8892(-6.35)	0.8777(-5.54)	0.8317(-1.53)

the rater role of users for recommendation. In most cases, the proposed framework with $\alpha = 0.001$ and $\beta = 0.001$ obtains much better performance than MF. These results demonstrate the importance of the rater role of users for recommendation.

- In general, with the increment of α (or β), the performance tends to first increases and then decreases. At certain regions, the performance is relatively stable. These patterns ease the parameter selection for the proposed framework in practice.
- Most of the time, better performance is achieved with relatively large values of α and small values of β. The map matrix $\mathbf{w}$ in Eq. (10) is learned from both observed item ratings and predicted item ratings, which are controlled by α and β, respectively. Since observed item ratings should be more reliable than predicted item ratings, relatively large values of α and small values of β can lead to the learned $\mathbf{w}$ more accurate.
- Compared to α, RecDual is more sensitive to β. A larger β tends to make the learned $\mathbf{w}$ overfit to predicted item ratings, which in turn leads to inaccurate estimates of $\mathbf{U}$ and $\mathbf{V}$.
- The parameters α and β have bigger impact on the proposed framework DualRec when the training data is sparse. When the training data is sparse, the proposed framework relies on the rater role of users more to learn $\mathbf{U}$ and $\mathbf{V}$. These observations further suggest the importance of the rater role of users for recommendation.

5. RELATED WORK

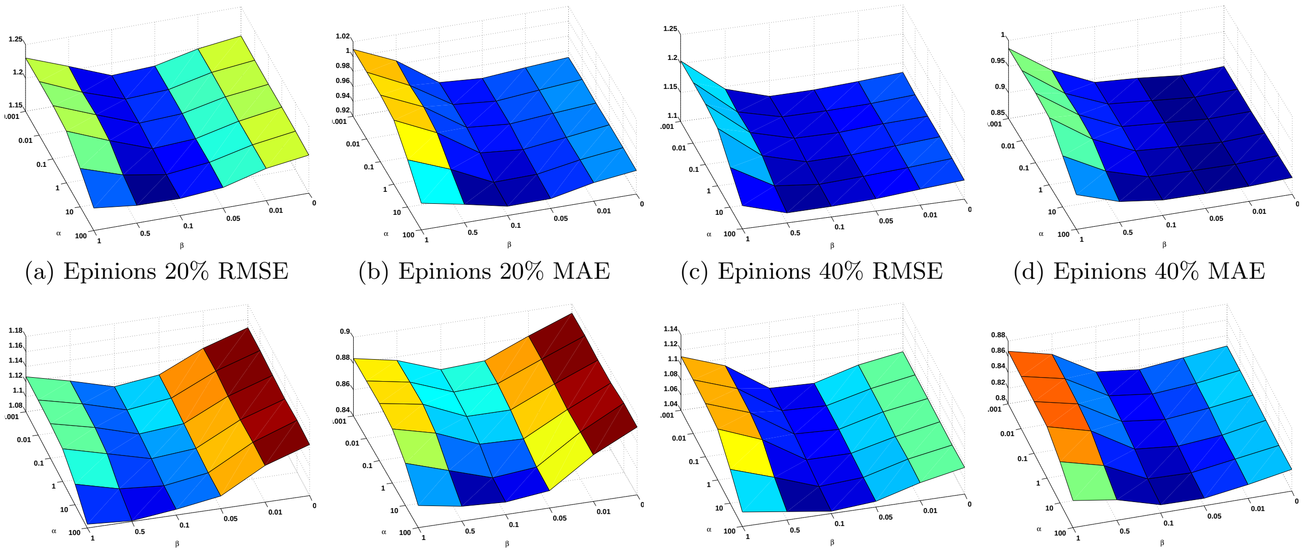
In this paper, we investigate the duel roles of users for recommendation; hence our work is related to traditional recommender systems that exploit the reviewer role of users from recommendation, as well as helpfulness rating prediction, which investigates the rater role of users.

5.1 Recommender Systems

Recommender systems can be roughly categorized as content based recommender systems[1, 28, 17, 20] and collaborative filtering[5, 23, 29]. Content based recommender systems recommend items based on user's profile information or similar to the ones the user preferred in the past, while CF only requires past user ratings to predict unknown ratings and has attracted more and more attention. Despite the success of CF based recommender systems, most of them only consider user's role as reviewers but ignore their role as raters[25, 16, 10, 4, 27, 2]. One challenge of traditional recommender system is that the rating matrix is usually very sparse, which degenerates the performance of many recommender systems. To deal with the data sparsity problem, there are some recommender systems relying on extra sources to find similar users or similar items. For example, [15, 14] utilize trust networks by assuming that a user's taste is similar to that of his/her trusted users; [4] incorporates user and item graphs to a matrix factorization framework, which captures user and item similarity; and [13] explores heterogeneous relations for collaborative filtering. Some algorithms also rely on building users profile[34, 19, 32]. For example, [34] uses initial interview process for cold-start users. [19] constructs tensor profiles for user/item pairs from their individual features. The proposed framework differs from aforementioned methods where we consider users' dual roles in real-world recommender systems. Helpfulness ratings encode users' opinions on reviews of items; hence they could be useful to infer users' opinions on items. Therefore, we propose to explicitly learn implicit item ratings from helpfulness ratings, which not only captures the rater role of users in recommendation but also has potentials to mitigate the data sparsity and cold-start problems in recommendation.

5.2 Helpfulness Rating Prediction

The rater role of users allows them to express their opinions on reviews about items via helpfulness ratings, and helpfulness rating prediction has attracted increasing atten-



(a) Epinions 20% RMSE (b) Epinions 20% MAE (c) Epinions 40% RMSE (d) Epinions 40% MAE

(e) Ciao 20% RMSE (f) Ciao 20% MAE (g) Ciao 40% RMSE (h) Ciao 40% MAE

Figure 4: Parameter Sensitivity of DualRec w.r.t to α and β

tion recently to understand the rater role of users. [3, 31, 11, 26, 12]. [3] estimates the helpfulness of product reviews by mining text and reviewer characteristics. [26] predicts helpfulness ratings by incorporating the context of reviews and social networks. [12] uses multilayer perceptron neural networks to predict online reviews. Despite various models proposed to predict helpfulness ratings, there's few work about using helpfulness ratings for recommendation. [21] uses helpfulness rating to measure the quality of the review and then uses the quality score as a weight to control the matrix factorization model. In other words, it uses helpfulness ratings of reviews to indicate the reliability about their associated item ratings; while the proposed framework explicitly models how to learn implicit item ratings from helpfulness ratings. Therefore, the proposed framework is substantially different from [21] - (1) the investigation perspectives are different; (2) the proposed formulations are different; and (3) the proposed framework has potentials to mitigate the data sparsity and cold-start problems by incorporating the rater role of users.

6. CONCLUSIONS

In this paper, we investigate users' dual roles, i.e., reviewers that give item ratings and raters that give helpfulness ratings, to advance traditional recommender systems since the vast majority of existing recommender systems only consider the reviewer role of users. To incorporate the rater role of users, we study how to learn implicit item ratings from helpfulness ratings and how to exploit the dual roles simultaneously for recommendation, which lead to a novel recommender system DualRec. Experimental results show that the proposed framework outperforms several state-of-the-art recommender systems. Further experiments are conducted to demonstrate the capability of the proposed framework in mitigating the data sparsity and cold-start problems for recommendation by incorporating the rater role of users.

There are several interesting directions that need further investigation. First, in this work, we choose matrix factorization as our basic model to incorporate the rater role of users, and we would like to investigate other basic models. For example, one new direction in recommender system is to explore implicit hierarchical structures of users and items by using deep non-negative matrix factorization[30]. We would investigate if user duel roles can be incorporated into that model. Second, for effectiveness, we choose linear regression to infer implicit item ratings from helpfulness ratings; we will investigate if other models can improve the performance. Since social networks are pervasively available in social media and provide independent sources for recommendation, we would like to investigate if social networks can be utilized to advance the implicit item rating learning process.

7. ACKNOWLEDGEMENTS

This material is based upon the work supported by, or in part by, Office of Naval Research (ONR) under grant number N000141410095 and Army Research Office (ARO) under grant number 025071.

8. REFERENCES

[1] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation, 1997.

[2] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.

[3] A. Ghose and P. G. Ipeirotis. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *Knowledge and Data Engineering, IEEE Transactions on*, 23(10):1498–1512, 2011.

[4] 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.

[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[6] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.

[7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[8] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[9] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[10] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, pages 30–37, 2009.

[11] N. Korfiatis, E. García-Bariocanal, and S. Sánchez-Alonso. Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content. *Electronic Commerce Research and Applications*, 11(3):205–217, 2012.

[12] S. Lee and J. Y. Choeh. Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046, 2014.

[13] C. Luo, W. Pang, Z. Wang, and C. Lin. Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In *2014 IEEE International Conference on Data Mining, ICDM*, pages 917–922, 2014.

[14] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[15] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.

[16] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[17] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[18] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust management*, pages 224–239. Springer, 2005.

[19] 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. ACM, 2009.

[20] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[21] S. Raghavan, S. Gunasekar, and J. Ghosh. Review quality aware collaborative filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 123–130. ACM, 2012.

[22] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[24] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *ICML*, volume 3, pages 704–711, 2003.

[25] N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization, 2005.

[26] J. Tang, H. Gao, X. Hu, and H. Liu. Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 1–8. ACM, 2013.

[27] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2712–2718. AAAI Press, 2013.

[28] R. Van Meteren and M. Van Someren. Using content-based filtering for recommendation, 2000.

[29] J. Wang, A. P. De Vries, and M. J. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006.

[30] S. Wang, J. Tang, and H. Liu. Exploring implicit hierarchical structures for recommender systems. In *Proceedings of 24th International Joint Conference on Artificial Intelligence*, 2015.

[31] W. Xiong and D. Litman. Automatically predicting peer-review helpfulness. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 502–507. Association for Computational Linguistics, 2011.

[32] M. Zhang, J. Tang, X. Zhang, and X. Xue. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval*, pages 73–82. ACM, 2014.

[33] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552. SIAM, 2006.

[34] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 315–324. ACM, 2011.