CrossFire: Cross Media Joint Friend and Item Recommendations

Kai Shu
Arizona State University
kai.shu@asu.edu

Suhang Wang
Arizona State University
suhang.wang@asu.edu

Jiliang Tang
Michigan State University
tangjili@cse.msu.edu

Yilin Wang
Arizona State University
ywang370@asu.edu

Huan Liu
Arizona State University
huan.liu@asu.edu

ABSTRACT

Friend and item recommendation on a social media site is an important task, which not only brings conveniences to users but also benefits platform providers. However, recommendation for newly launched social media sites is challenging because they often lack user historical data and encounter data sparsity and cold-start problem. Thus, it is important to exploit auxiliary information to help improve recommendation performances on these sites. Existing approaches try to utilize the knowledge transferred from other mature sites, which often require overlapped users or similar items to ensure an effective knowledge transfer. However, these assumptions may not hold in practice because 1) Overlapped user set is often unavailable and costly to identify due to the heterogeneous user profile, content and network data, and 2) Different schemes to show item attributes across sites cause the attribute values inconsistent, incomplete, and noisy. Thus, how to transfer knowledge when no direct bridge is given between two social media sites remains a challenge. In addition, another auxiliary information we can exploit is the mutual benefit between social relationships and rating preferences within the platform. User-user relationships are widely used as side information to improve item recommendation, whereas how to exploit user-item interactions for friend recommendation is rather limited. To tackle the aforementioned challenges, we propose a *Cross* media joint *F*riend and *I*tem *Re*commendation framework (*CrossFire*), which can capture both 1) cross-platform knowledge transfer, and 2) within-platform correlations among user-user relations and user-item interactions. Empirical results on real-world datasets demonstrate the effectiveness of the proposed framework.

KEYWORDS

Cross media recommendation; joint learning; data mining

1 INTRODUCTION

Social media websites provide users with multiple services such as online reviews, networking, social publishing, etc. To improve personalized services, social media sites often attempt to suggest potential information that will match the interests of users or potential friends that users will form relationships with. Recommender systems, which aim to solve the aforementioned problems, are attracting more and more attention in recent years [20, 21, 27]. For those mature social media sites, they have abundant historical information to help build recommender system, whereas those newly launched sites often lack these information and encounter the data sparsity and cold-start challenges [22]. To build effective and practical friend and item recommendation systems for the newly launched sites, it's natural and necessary to explore *auxiliary information* from different aspects.

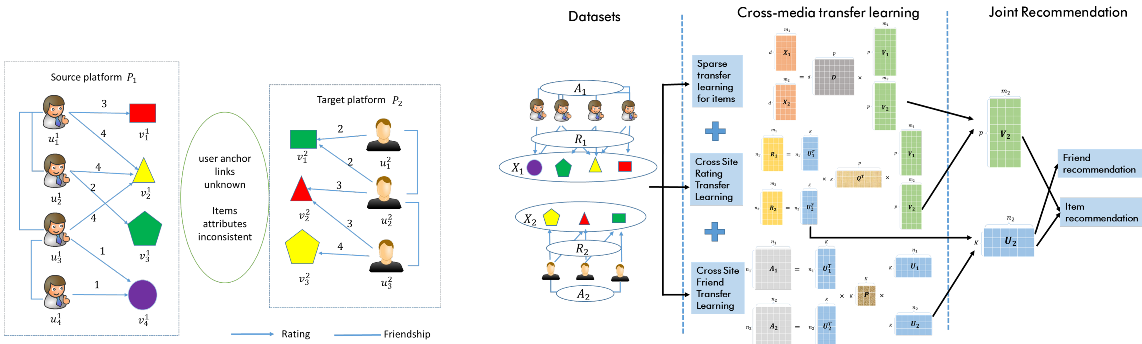
One popular way of exploring auxiliary information is to transfer the knowledge from the mature platform to newly created platform by assuming that either (i) there are anchor links between users across two platforms and thus knowledge can be transferred through these anchor links [3, 13]; or (ii) items attributes are consistent and thus we can directly utilize similarity between items to transfer the knowledge [11]. However, in practice, these assumptions may not hold. A typical scenario is shown in Figure 1(a), where P_1 is a mature platform and P_2 is a newly launched one. Users in P_1 can form a social network and give ratings to items in P_1. Similarly, users in P_2 can also form links and rate items in P_2. However, there are no anchor links between users in P_1 and users in P_2 and thus no information can be directly transferred between users in P_1 and P_2. In addition, directly measuring the similarity between items in P_1 and P_2 may not be applicable as different sites encode items in different schemes and result in inconsistent attribute fields and many missing values. The majority of existing work that exploits anchor links or item similarities cannot be directly applied. Therefore, it is important to study the novel and challenging problem of cross platform recommendation when no direct bridge between users or items is given.

In addition to transferring knowledge across platforms, we can also exploit auxiliary information within a single platform. As shown in Figure 1(a), on a social media site, users can usually form relationships with others as well as express their preferences to items. For example, in GoodReads[1], people can follow and be followed by other users and also give ratings to the books they read. According to social correlation theories, users' preferences towards items and friends have inseparable correlations. Homophily [18] theory shows that users who have similar attributes are more likely to become friends [18]. Likewise, social influence theory [17] suggests that users are more likely to be influenced by their friends and express similar ratings to those items visited by their friends.

[1]https://www.goodreads.com/

WSDM 2018, Marina Del Rey, CA, USA

DOI: 10.1145/3159652.3159692



(a) An example of cross social media joint recommendations

(b) The flowchart of the proposed framework CrossFire

Figure 1: An illustration of the problem scenario for cross-media joint friend and item recommendation, and the basic idea of proposed framework integrating within-platform correlations and cross-platform information for joint recommendations.

However, even though much efforts have been devoted to exploiting user-user relations to help item recommendations, how to exploit user-item interactions for friend recommendation remains limited [23]. Previous works also show that a positive correlation between users' interests and social relationships [26], which may indicate an added value to utilize user interests to help friend recommendations in social media. Thus, we can jointly recommend items and friends by exploiting the correlations among them. As shown in Figure 1(a), suppose u_1^2 and u_2^2 are friends, user u_1^2 is likely to be influenced by his/her friend u_2^2 and gives a similar rating to item v_1^2. Also, user u_2^2 and u_3^2 both give a rating score 3 to item v_2^2, so they are more likely to form a relationship.

Cross-platform knowledge transfer and within-platform joint learning have complementary information, which both provide new perspectives to help improve the recommendation performances for the newly launched platform. In this paper, we investigate: (1) how to transfer the information from source platform to target platform when there is no straightforward way to bridge two platforms, and (2) how to mathematically formulate joint friend and item recommendation so as to improve recommendation performance. To tackle these challenges, we propose a novel *Cross*-media joint *F*riend and *I*tem *Re*commendation framework (*CrossFire*). The framework can i) build the implicit bridge to transfer knowledge by utilizing the observation that the user behaviors share similar patterns across different networks, and the assumption that item features on two platforms share the same dictionary; ii) learn user and item latent features by exploiting the correlation between user-user relations and user-item interactions to perform joint friend and item recommendations. The main contributions are as below:

- We study a novel problem of cross-media joint friend and item recommendations when no anchor links are available;
- We propose a new framework CrossFire which integrates within-platform correlations and cross-media information into a coherent model for joint friend and item recommendations, and an optimization algorithm to solve it;
- We conduct experiments on real-world social media sites to demonstrate the effectiveness of the proposed framework CrossFire for friend and item recommendations.

2 PROBLEM FORMULATION

We first introduce the notations of this paper, and then give the formal problem definition. Let $\mathcal{U}_1 = \{u_1^1, u_2^1, \ldots, u_{n_1}^1\}$ and $\mathcal{V}_1 = \{v_1^1, v_2^1, \ldots, v_{m_1}^1\}$ be the sets of users and items in the source social media site, where n_1 and m_1 are the numbers of users and items, respectively. $\mathcal{U}_2 = \{u_1^2, u_2^2, \ldots, u_{n_2}^2\}$ and $\mathcal{V}_2 = \{v_1^2, v_2^2, \ldots, v_{m_2}^2\}$ denote the sets of users and items in the target social media site where n_2 and m_2 denote the number of users and items, respectively. We also use $\mathbf{X}_1 \in \mathbb{R}^{d \times m_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{d \times m_2}$ to denote the item features in source and target domains, separately; where d is the dimension of item feature vectors. On each social media site, users can rate the items and we use $\mathbf{R_1} \in \mathbb{R}^{n_1 \times m_1}$ and $\mathbf{R}_2 \in \mathbb{R}^{n_2 \times m_2}$ to denote the user-item rating matrices for the source domain and target domain, respectively. Users can become friend with other users and we use $\mathbf{A}_1 \in \{1, 0\}^{n_1 \times n_1}$ and $\mathbf{A}_2 \in \{1, 0\}^{n_2 \times n_2}$ to denote the user-user adjacency matrices on the source and target social media site, respectively. A very common situation is that the source domain is a mature source media site while the target site is newly launched. Thus, the rating matrix $\mathbf{R}_1$ and user-user adjacency matrix $\mathbf{A}_1$ in the source domain are relatively dense while $\mathbf{R}_2$ and $\mathbf{A}_2$ in the target domain are very sparse. Much work has demonstrated that we can learn better user and item latent features with dense rating and user-user matrices for friend and item recommendations while it is very difficult to make reasonable recommendations with very sparse rating matrices. Thus, we want to use $\mathbf{R}_1$ and $\mathbf{A}_1$ in the source domain to help friend and item recommendations in the target domain. We assume that there is no explicit correspondence information among users and items. With the notations given above, the problem is formally given as,

Given rating matrix $\mathbf{R}_1$*, user-user matrix* $\mathbf{A}_1$ *and item-feature matrix* $\mathbf{X}_1$ *in source social media site, rating matrix* $\mathbf{R}_2$*, user-user link matrix* $\mathbf{A}_2$ *and item-feature matrix* $\mathbf{X}_2$ *in target social media site, we aim to make friend and item recommendations on target social media site.*

3 CROSS MEDIA JOINT RECOMMENDATIONS

In this section, we present the details of proposed framework for cross-media joint friend and item recommendations. As shown in Figure 1(b), our framework includes three major parts. We first introduce how to exploit sparse transfer learning method to model cross-media item information. Then we propose how to perform cross site rating transfer learning. Finally, we present cross site friend transfer learning followed by the proposed framework.

3.1 Sparse Transfer Learning for Items

Different online social network sites have different structures and schemes to present item detail attributes, which causes the inconsistent attribute fields and many missing values. Thus, it becomes ineffective to bridge items across sites directly using similarity metrics, such as distance-based and frequency-based [24] on these attributes. To make the bridge between items on two social network sites, we propose to utilize sparse learning based transfer learning. Specifically, the source feature matrix $\mathbf{X}_1$ can be reconstructed as $\mathbf{X}_1 \approx \mathbf{D}\mathbf{V}_1$ and the target matrix $\mathbf{X}_2$ is reconstructed as $\mathbf{X}_2 \approx \mathbf{D}\mathbf{V}_2$, where $\mathbf{D}$ is the dictionary shared by source and target, $\mathbf{V}_1 \in \mathbb{R}^{p \times m_1}$ and $\mathbf{V}_2 \in \mathbb{R}^{p \times m_2}$ are the sparse representations for $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. The essential idea is that $\mathbf{X}_1$ and $\mathbf{X}_2$ share a dictionary $\mathbf{D}$, which behaviors as a bridge to transfer knowledge from $\mathbf{X}_1$ to $\mathbf{X}_2$. With this assumption, the sparse learning objective is given as:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{V}_1,\mathbf{V}_2} \; & \|\mathbf{X}_1 - \mathbf{D}\mathbf{V}_1\|_F^2 + \|\mathbf{X}_2 - \mathbf{D}\mathbf{V}_2\|_F^2 + \gamma(\|\mathbf{V}_1\|_1 + \|\mathbf{V}_2\|_1) \\ & s.t. \; \|\mathbf{d}_j\|_2^2 \leq 1, j = 1, \ldots, p \end{aligned} \tag{1}$$

where $\mathbf{d}_j$ is the j_{th} column of $\mathbf{D}$.

To make the dictionary matrix $\mathbf{D}$ satisfy the intrinsic geometric structure of the item features, we incorporate a Graph Regularized Sparse Coding (GraphSC) method [41]. The basic assumption of GraphSC is that if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the intrinsic geometry of data distributions, their codings $\mathbf{v}_i$ and $\mathbf{v}_j$ are also close. Thus, given $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2] \in \mathbb{R}^{d \times (m_1 + m_2)}$, GraphSC constructs a K-nearest neighbor graph G with $(m_1 + m_2)$ nodes representing all data points. Let $\mathbf{H}$ be the weight matrix of G; if $\mathbf{x}_i$ is among the K-nearest neighbors with $\mathbf{x}_j$, then $\mathbf{H}_{ij} = \exp^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma}}$, where σ is the scalar to control the bandwidth; otherwise, $\mathbf{H}_{ij} = 0$. Then the graph regularization term tries to minimize

$$\frac{1}{2} \sum_{i,j=1}^{m_1+m_2} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \mathbf{H}_{ij} = Tr(\mathbf{V}\mathbf{L}\mathbf{V}^T) \tag{2}$$

where $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]$. $\mathbf{L} = \mathbf{S} - \mathbf{H}$ is the Laplacian matrix and $\mathbf{S}$ is a diagonal matrix with the diagonal element $\mathbf{S}_{ii} = \sum_{j=1}^{m_1+m_2} \mathbf{H}_{ij}$. Moreover, to enforce unified codings for both domains, we also regularize the model with an additional term called maximum mean discrepancy regularization [12] as follows,

$$MMD = \|\frac{1}{m_1}\mathbf{V}_1\mathbf{1} - \frac{1}{m_2}\mathbf{V}_2\mathbf{1}\|_2^2 = Tr(\mathbf{V}\mathbf{M}\mathbf{V}^T) \tag{3}$$

which is the ℓ_2-norm of the difference between mean samples of the source and target domains in the sparse coding space. $\mathbf{M}$ is the MMD matrix and is computed as $\mathbf{M}_{ij} = 1/m_1^2$ if $v_i, v_j \in \mathcal{V}_1$, $\mathbf{M}_{ij} = 1/m_2^2$ if $v_i, v_j \in \mathcal{V}_2$ and $\frac{-1}{m_1 m_2}$ otherwise.

The graph regularization term in Eq.(2) and the MMD regularization term in Eq.(3) together guide the learning process of $\mathbf{D}$ and $\mathbf{V}$ so that the probability distribution of both domains are drawn close under the new representation $\mathbf{V}$. Thus, we can formulate the sparse transfer learning for items as,

$$\begin{aligned} \min_{\mathbf{V}_1,\mathbf{V}_2,\mathbf{D}} & \| \mathbf{X}_1 - \mathbf{D}\mathbf{V}_1\|_F^2 + \|\mathbf{X}_2 - \mathbf{D}\mathbf{V}_2\|_F^2 + \gamma(\|\mathbf{V}_1\|_1 + \|\mathbf{V}_2\|_1) \\ & + Tr(\mathbf{V}(\mu\mathbf{L} + \nu\mathbf{M})\mathbf{V}^T), \quad s.t. \|\mathbf{d}^i\|_2^2 \leq 1, i = 1, \ldots, p \end{aligned} \tag{4}$$

where μ and ν are used as a trade-off for the contributions of graph regularization term and MMD, γ is to control the level of sparsity.

3.2 Cross Site Rating Transfer Learning

Following existing work that assume users' rating behaviors share common patterns across sites [9], we propose a transfer learning model to better learn user and item latent representations. With the sparse representations of items for both the source and target domains, we introduce another projection matrix $\mathbf{Q}^{p \times K}$, which projects the sparse representations to the latent item feature representations, i.e., $\mathbf{Q}^T\mathbf{V}_1$ and $\mathbf{Q}^T\mathbf{V}_2$. Thus, we can formulate the objective function of cross-media rating transfer learning as below,

$$\begin{aligned} \min_{\mathbf{U}_1,\mathbf{V}_1\mathbf{U}_2,\mathbf{V}_2,\mathbf{Q}} & \|\mathbf{W}_1 \odot (\mathbf{R}_1 - \mathbf{U}_1{}^T\mathbf{Q}^T\mathbf{V}_1)\|_F^2 + \|\mathbf{W}_2 \odot (\mathbf{R}_2 - \mathbf{U}_2{}^T\mathbf{Q}^T\mathbf{V}_2)\|_F^2 \\ & s.t. \; \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \end{aligned} \tag{5}$$

where $\odot$ denotes the Hadamard product and $\mathbf{W}_1$ ($\mathbf{W}_2$) controls the contribution of $\mathbf{R}_1$ ($\mathbf{R}_2$). $\mathbf{U}_1 \in \mathcal{R}^{K \times n_1}$ ($\mathbf{U}_2 \in \mathcal{R}^{K \times n_2}$) is the latent feature representations in source (target) domain. The orthogonal constraint on $\mathbf{Q}$ is to ensure that each column of $\mathbf{Q}$ are independent.

3.3 Cross Site Friend Transfer Learning

Existing research on cross social network analysis has demonstrated that different social network platforms may have similar network structures and characteristics, e.g., the number of user relations falls into power-law distributions [2]. Following traditional setting, we decompose the user-user link matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ to map them to a shared latent space and obtain user latent features $\mathbf{U}_1$ and $\mathbf{U}_2$. In order to model the latent features shared across different social media sites, we also exploit a shared interaction matrix factorization $\mathbf{P}$. Then the modeling formulation is,

$$\min_{\mathbf{U}_1,\mathbf{U}_2,\mathbf{P}} \|\mathbf{Y}_1 \odot (\mathbf{A}_1 - \mathbf{U}_1{}^T\mathbf{P}\mathbf{U}_1)\|_F^2 + \|\mathbf{Y}_2 \odot (\mathbf{A}_2 - \mathbf{U}_2{}^T\mathbf{P}\mathbf{U}_2\|_F^2 \tag{6}$$

where $\odot$ denotes the Hadamard product and $\mathbf{Y}_1$ ($\mathbf{Y}_2$) controls the contribution of $\mathbf{A}_1$ ($\mathbf{A}_2$). The essential idea of using shared $\mathbf{P}$ is that: $\mathbf{P}$ is the interaction matrix which captures the interaction property of users, i.e., the connection status of u_s^i and u_t^i is represented via the interaction matrix $\mathbf{P}$ as $\mathbf{U}_i(s,:)^T\mathbf{P}\mathbf{U}_j(:,t)$. Since users connection behaviors are consistent in different sites, e.g., the structure of social networks are similar, $\mathbf{P}$ should also be similar across sites.

3.4 Proposed Framework

In this section, we combine the three components together, and present the framework of cross-media joint friend and item recommendations named CrossFire. CrossFire aims to solve following optimization problem,

$$\min_{\theta} \underbrace{\sum_{i=1}^{2} \|\mathbf{X}_i - \mathbf{D}\mathbf{V}_i\|_F^2 + \gamma\|\mathbf{V}_i\|_1 + Tr(\mathbf{V}(\mu\mathbf{L} + \nu\mathbf{M})\mathbf{V}^T)}_{\text{Sparse Transfer Learning}}$$
$$+\alpha \underbrace{\sum_{i=1}^{2} \|\mathbf{W}_i \odot (\mathbf{R}_i - \mathbf{U}_i^T\mathbf{Q}^T\mathbf{V}_i)\|_F^2}_{\text{Cross Site Item Recommendation}} + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) \tag{7}$$
$$+\beta \underbrace{\sum_{i=1}^{2} \|\mathbf{Y}_i \odot (\mathbf{A}_i - \mathbf{U}_i^T\mathbf{P}\mathbf{U}_i)\|_F^2}_{\text{Cross Site Friend Recommendation}} + \lambda\sum_{i=1}^{2}\|\mathbf{U}_i\|_F^2$$
$$s.t.\ \|\mathbf{d}_j\|_2^2 \le 1, j = 1, \ldots, p,\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$$

where the first part is to perform item sparse transfer learning; the second part captures the cross-media rating transfer learning; the third term models the user relations transfer learning. By incorporating these components together, we are able to make joint recommendations for items and friends simultaneously with the resultant latent features.

4 AN OPTIMIZATION FRAMEWORK

In this section, we present the details of the optimization process for the proposed framework CrossFire. If we update the variables jointly, the objective function in Eq. 7 is not convex. Thus, we use alternating least square method to iteratively optimize each variable separately. Next, we will introduce the updating rules. For simplicity, we use $\mathcal{L}$ to denote the objective function in Eq. 7.

4.1 Update Rules

In this section, we will introduce the updating rules for each variable in details.

4.1.1 Update $\mathbf{D}$. The objective function related to $\mathbf{D}$ can be rewrote as,

$$\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{D}\mathbf{V}\|_F^2 \quad s.t.\ \|\mathbf{d}_j\|_2^2 \le 1, j = 1, \ldots, p \tag{8}$$

where $\mathbf{X} = [\mathbf{X}_1,\ \mathbf{X}_2]$ and $\mathbf{V} = [\mathbf{V}_1,\ \mathbf{V}_2]$. Eq.(8) is a standard dictionary learning problem and can be solved using the algorithm proposed in [8].

4.1.2 Update $\mathbf{V}$. Since updating $\mathbf{V}$ involves the l_1 norm, we propose to use Alternating Direction Method of Multiplier (ADMM) [1, 38] to update $\mathbf{V}$. By introducing an auxiliary variable $\mathbf{Z} = \mathbf{V}$ and et $\tilde{\mathbf{L}} = \mu\mathbf{L} + \nu\mathbf{M}$, we can rewrite the objective function as follows,

$$\min_{\mathbf{V}} \underbrace{\|\mathbf{X} - \mathbf{D}\mathbf{V}\|_F^2 + Tr(\mathbf{V}\tilde{\mathbf{L}}\mathbf{V}^T) + \alpha\sum_{i=1}^{2}\|\mathbf{W}_i \odot (\mathbf{R}_i - \mathbf{U}_i^T\mathbf{Q}^T\mathbf{V}_i)\|_F^2}_{g(\mathbf{V})} + \underbrace{\gamma\|\mathbf{Z}\|_1}_{h(\mathbf{Z})}$$
$$s.t. \mathbf{V} - \mathbf{Z} = \mathbf{0} \tag{9}$$

This is a standard l_1 regularized ADMM problem [1]. The updating function from step t to step $t+1$ is,

$$\mathbf{V}^{t+1} := \text{argmin}(g(\mathbf{V}) + \rho/2\|\mathbf{V} - \mathbf{Z}^t + \mathbf{E}^t\|_2^2) \tag{10}$$
$$\mathbf{Z}^{t+1} := \mathcal{T}_{\gamma/\rho}(\mathbf{V}^{t+1} + \mathbf{E}^t) \tag{11}$$
$$\mathbf{E}^{t+1} := \mathbf{E}^t + \mathbf{V}^{t+1} - \mathbf{Z}^{t+1} \tag{12}$$

where ρ is the trade-off parameter and $\mathcal{T}_{\gamma/\rho}(\mathbf{V})$ is a the proximal function for l_1 norm (i.e. soft-thresholding operator) [19] defined as follows,

$$\left[\mathcal{T}_{\gamma/\rho}(\mathbf{V})\right]_{ij} = \text{sign}(\mathbf{V}_{ij})(|\mathbf{V}_{ij}| - \gamma/\rho)_+ \tag{13}$$

To solve Eq. 10, we use gradient descent method to update $\mathbf{V}$ as in Algorithm 1. The partial derivative of updating $\mathbf{V}$ is,

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = 2\mathbf{D}^T(\mathbf{D}\mathbf{V} - \mathbf{X}) + \mathbf{V}\tilde{\mathbf{L}} + \rho(\mathbf{V} - \mathbf{Z} + \mathbf{E})$$
$$+2\alpha\left[\mathbf{Q}\mathbf{U}_1[\mathbf{W}_1 \odot (\mathbf{U}_1^T\mathbf{Q}^T\mathbf{V}_1 - \mathbf{R}_1)], \mathbf{Q}\mathbf{U}_2[\mathbf{W}_2 \odot (\mathbf{U}_2^T\mathbf{Q}^T\mathbf{V}_2 - \mathbf{R}_2)]\right]$$

Algorithm 1 Update $\mathbf{V}$

Require: Initial feasible $\mathbf{V}$, $\mathbf{Z}$, $\mathbf{E}$, $\rho = 0.5$, γ, maxsteps
Ensure: Updated $\mathbf{V}$
1: **for** t = 1 to maxsteps **do**
2: Update $\mathbf{V}^{t+1}$ using $\mathbf{V}^{t+1} = \mathbf{V}^t - \epsilon\frac{\partial\mathcal{L}}{\partial\mathbf{V}}$ via Eq. 14
3: Update $\mathbf{Z}^{t+1}$ via Eq. 11
4: Update $\mathbf{E}^{t+1}$ via Eq. 12
5: **end for**
6: Return $\mathbf{V}$

4.1.3 Update $\mathbf{U}_i$ *and* $\mathbf{P}$. The partial derivative of the objective function w.r.t $\mathbf{U}_i$ is given as

$$\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\mathbf{U}_i} = \alpha\mathbf{Q}^T\mathbf{V}_i[\mathbf{W}_i \odot (\mathbf{U}_i^T\mathbf{Q}\mathbf{V}_i - \mathbf{R}_i)]^T + \lambda\mathbf{U}_i$$
$$+\beta\mathbf{P}^T\mathbf{U}_i[\mathbf{Y}_i \odot (\mathbf{U}_i^T\mathbf{P}\mathbf{U}_i - \mathbf{A}_i)] + \beta\mathbf{P}\mathbf{U}_i[\mathbf{Y}_i \odot (\mathbf{U}_i^T\mathbf{P}\mathbf{U}_i - \mathbf{A}_i)]^T \tag{14}$$

and the partial derivative of the objective function w.r.t $\mathbf{P}$ is

$$\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\mathbf{P}} = \beta\sum_{i=1}^{2}[\mathbf{U}_i(\mathbf{Y}_i \odot \mathbf{U}_i^T\mathbf{P}\mathbf{U}_i)\mathbf{U}_i^T - \mathbf{U}_i(\mathbf{Y}_i \odot \mathbf{A}_i)\mathbf{U}_i^T] + \lambda\mathbf{P} \tag{15}$$

4.1.4 Update $\mathbf{Q}$. The objective with respect to $\mathbf{Q}$ is as follows,

$$\min_{\mathbf{Q}} \alpha\sum_{i=1}^{2}\|\mathbf{W}_i \odot (\mathbf{R}_i - \mathbf{U}_i^T\mathbf{Q}^T\mathbf{V}_i)\|_F^2 + \lambda\|\mathbf{Q}\|_F^2,\ s.t.\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \tag{16}$$

We use a gradient descent optimization procedure with curvilinear search [35] to solve it. The gradient can be calculated as,

$$\mathbf{G} = \frac{\partial\mathcal{L}}{\partial\mathbf{Q}} = 2\alpha\sum_{i=1}^{2}\mathbf{V}_i[\mathbf{W}_i \odot (\mathbf{U}_i^T\mathbf{Q}^T\mathbf{V}_i - \mathbf{R}_i)]^T\mathbf{U}_i^T + 2\lambda\mathbf{Q} \tag{17}$$

We then defined $\mathbf{F} \in \mathbb{R}^{K\times K}$ as $\mathbf{F} = \mathbf{G}\mathbf{Q}^T - \mathbf{Q}\mathbf{G}^T$. Note that $\mathbf{F}^T = -\mathbf{F}$ and thus $\mathbf{F}$ is skew-symmetric. The next new point can be searched as a curvilinear function of a step size variable τ such that,

$$\mathbf{S}(\tau) = (\mathbf{I} + \frac{\tau}{2}\mathbf{F})^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{F})\mathbf{Q} \tag{18}$$

Algorithm 2 Update Q

Require: Initial feasible $\mathbf{Q}$, $0 < \mu < 1, 0 < \rho_1 < \rho_2 < 1$
Ensure: Updated $\mathbf{Q}$
Compute $\mathbf{F}$, $\mathbf{G}$, $\mathcal{L}_{\tau}^{'}(\mathbf{S}(0))$ respectively; set $\tau = 1$
for s = 1 to maxsteps **do**
  Compute $\mathbf{S}(\tau)$ via Eq.(18), $\mathcal{L}_{\tau}^{'}(\mathbf{S}(\tau))$ via Eq.(20)
  if Armijio-Wolfe conditions are satisfied **then** break-out
  end if
  $\tau = \mu\tau$
end for
Update $\mathbf{Q}$ as $\mathbf{Q} = \mathbf{S}$
Return $\mathbf{Q}$

Algorithm 3 The optimization process of CrossFire framework

Require: $\{\mathbf{X}_i, \mathbf{R}_i, \mathbf{A}_i\}_{i=1,2}$, $\alpha, \beta, \gamma, \lambda, \mu, \nu$
Ensure: $\{\mathbf{U}_i, \mathbf{V}_i\}_{i=1,2}$, $\mathbf{P}, \mathbf{Q}, \mathbf{D}$
1: Initialize $\{\mathbf{U}_i, \mathbf{V}_i\}_{i=1,2}$, $\mathbf{P}, \mathbf{Q}, \mathbf{D}$
2: Precompute $\mathbf{L}$, $\mathbf{M}$ and $\tilde{\mathbf{L}} = \mu\mathbf{L} + \nu\mathbf{M}$
3: **repeat**
4:   Update $\mathbf{D}$ via algorithm proposed in [8]
5:   Update $\mathbf{V}_1$, $\mathbf{V}_2$ with Algorithm 1
6:   Update $\mathbf{U}_1$, $\mathbf{U}_2$ as $\mathbf{U}_i \leftarrow \mathbf{U}_i - \epsilon\frac{\partial\mathcal{L}}{\partial\mathbf{U}_i}$ using Eq. 14
7:   Update $\mathbf{P}$ as $\mathbf{P} \leftarrow \mathbf{P} - \epsilon\frac{\partial\mathcal{L}}{\partial\mathbf{P}}$ using Eq. 15
8:   Update $\mathbf{Q}$ via Algorithm 2
9: **until** Convergence

It can be proved that $\mathbf{S}(\tau)$ is orthogonal based on Cayley transformation [6]. Thus we can stay in the feasible region along the curve defined by τ. We determine a proper step size τ satisfying the following Armijo-Wolfe conditions,

$$\mathcal{L}(\mathbf{S}(\tau)) \leq \mathcal{L}(\mathbf{S}(0)) + \rho_1\tau\mathcal{L}_{\tau}'(\mathbf{S}(0)),\ \mathcal{L}_{\tau}'(\mathbf{S}(\tau)) \geq \rho_2\mathcal{L}_{\tau}'(\mathbf{S}(0)) \quad (19)$$

Here $\mathcal{L}_{\tau}^{'}(\mathbf{S}(\tau))$ is the derivative of $\mathcal{L}$ w.r.t τ,

$$\mathcal{L}_{\tau}^{'}(\mathbf{S}(\tau)) = -Tr(\mathbf{R}(\tau)^T(\mathbf{I} + \frac{\tau}{2}\mathbf{F})^{-1}\mathbf{F}\frac{\mathbf{Q} + \mathbf{S}(\tau)}{2}) \quad (20)$$

where $\mathbf{R}(\tau) = \nabla_{\mathbf{S}_{\tau}}\mathcal{L}(\mathbf{S}(\tau))$. Obviously, $\mathbf{S}(0) = \mathbf{Q}$ and thus $\mathbf{R}(0) = \nabla_{\mathbf{Q}}\mathcal{L}(\mathbf{Q}) = \mathbf{G}$. Therefore $\mathcal{L}_{\tau}^{'}(\mathbf{S}(0)) = -\frac{1}{2}\|\mathbf{F}\|_F^2$. Details of updating $\mathbf{Q}$ is shown in Algorithm 2.

4.2 Algorithm of CrossFire

To this end, we give the detailed algorithm to learn the parameters for CrossFire in Algorithm 3. In line 1, we initialize the parameters $\{\mathbf{U}_i, \mathbf{V}_i\}_{i=1,2}$, $\mathbf{P}, \mathbf{Q}$ and $\mathbf{D}$. In line 2, we precompute graph laplacian matrix $\mathbf{L}$ and MMD matrix $\mathbf{M}$. Next, we update these parameters sequentially from Line 4 to Line 8 until convergence. Note that ϵ is the learning rate for each iteration step. Finally, based on the resultant latent matrix representations of users and items, we can use them to perform friend and item recommendation tasks.

The convergence of the algorithm is guaranteed. The reason is that we use gradient descent to update the parameters iteratively, and the objective value will monotonically reduce. Note that the objective function in Eq. 7 is non-negative, so the proposed algorithm will converge and it will achieve a local optimal value.

Table 1: The statistics of datasets

Dataset	Book		Movie	
	Source	Target	Source	Target
Platform	GoodReads	BookLikes	Epinions	Ciao
# users	7,490	3,853	5,588	2,126
# items	6,946	5,884	8,072	2,426
# ratings	199,915	134,525	109,804	24,012
# user links	120,790	96,327	215,916	43,362

4.3 Time Complexity

For the time complexity of proposed algorithm, we mainly focus on the parameter learning process. For parameter $\mathbf{D}$, we adopt the method that uses a Lagrange dual which has been shown to be much efficient [8]. The computation cost is approximately $O(d(m_1+m_2)p)$. Considering that we use ADMM to update $\mathbf{V}$ in Algorithm 1, the major cost is to update $\mathbf{V}$ and the cost is about $O(t(K(m_1 + m_2)p + K(p+1)(m_1n_1+m_2n_2)+dp^2(m_1+m_2)+p(m_1+m_2)^2))$, where t is the number of iteration steps for updating $\mathbf{V}$. The cost of updating $\mathbf{U}_i$ is $O(Kpm_i + K^2n_i + Kn_i^2 + Kn_im_i)$. Similarly, the cost of updating $\mathbf{P}$ is $O(K(n_1^2+n_2^2)+K^2(n_1+n_2))$. At last, $\mathbf{Q}$ is updated using Algorithm 8 and the computation cost is approximately $O(Kp(n_1 + n_2) + (p + K)(n_1m_1 + n_2m_2) + K^2(m_1 + m_2))$.

5 EXPERIMENT EVALUATION

In this section, we will conduct experiments on real-world datasets to demonstrate the effectiveness of the proposed framework. Specifically, we aim to answer the following research questions:

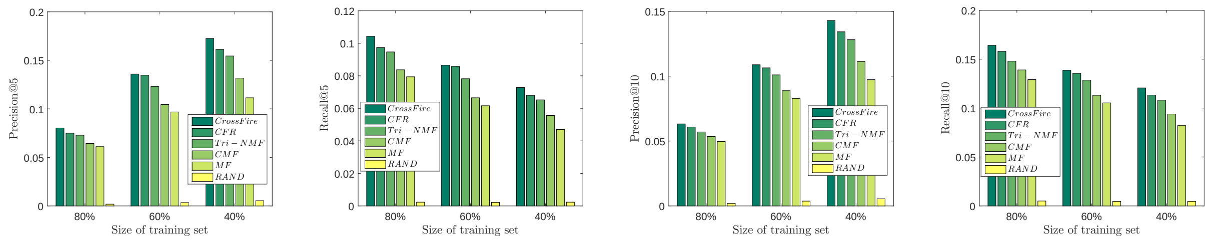
- Is CrossFire able to improve friend and item recommendation by exploiting within-platform correlations and cross-platform transferring information simultaneously?
- How effective are cross-media learning and joint friend and item prediction, respectively, in improving the recommendation performance of CrossFire?

To answer the first question, we compare the performance of friend and item recommendations of CrossFire with the state-of-the-art friend and item recommender systems, respectively. We then investigate the effects of cross-media recommendation and joint prediction on the proposed framework by doing parameter analysis to answer the second question.

5.1 Datasets

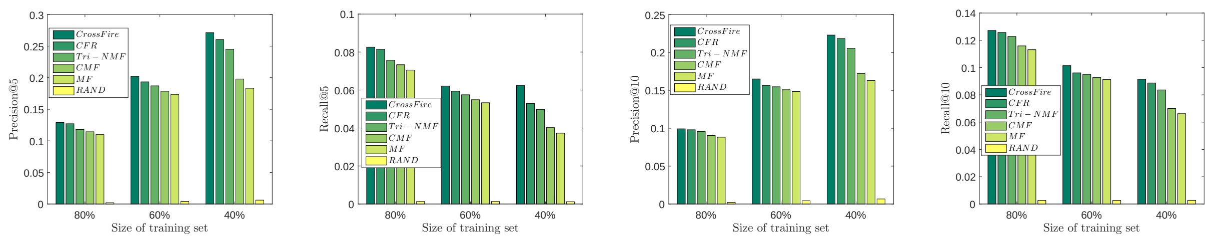
We ensure each site has the following information: *user-item interactions*, *user-user relations*, and *item features*. As shown in Table 1, we have two pairs of cross-media social media datasets to evaluate the proposed framework, i.e., *Book* and *Movie*. The *Book* data is collected from two book review social media sites, GoodReads[2] and BookLikes[3], using web crawlers from April 2017 to May 2017[4]. Users on GoodReads and BookLikes can rate the books they read of score 1 to 5 and they can follow and be followed by others. The *Movie* dataset includes two item review sites Epinions and Ciao

[2]https://www.goodreads.com
[3]http://booklikes.com/
[4]The dataset will be publicly available in the first author's homepage



(a) Precision@5 (b) Recall@5 (c) Precision@10 (d) Recall@10

Figure 2: Precision@5, Recall@5, Precision@10 and Recall@10 on *Ciao*.



(a) Precision@5 (b) Recall@5 (c) Precision@10 (d) Recall@10

Figure 3: Precision@5, Recall@5, Precision@10 and Recall@10 on *BookLikes*.

and public available [5]. We only keep items that belong to "movie" category. Users' ratings range from 1 to 5 and they can also establish social relations (trust relations). Since different schemes to show item attributes cause the attribute values inconsistent and incomplete, we only assume the most common text information for items, i.e., *name* and *description*, are available. Thus, Item features are represented using Bag-of-words model on the text. In the sense, it may not be applicable to directly compute similarity score on the attribute values since most of them are not available. Note that we select the source and target platform for each cross-media data based on real-world popularities of those sites.

5.2 Friend Recommendation

In this subsection, we check whether the proposed framework CrossFire can improve the performance of friend recommendation.

5.2.1 Experimental Settings. We randomly choose a fraction of $x\%$ positive user-user pairs for training and use the remaining $1-x\%$ of all links for testing, where x is varied in $\{80, 60, 40\}$. We use top-k evaluation metrics to measure the recommendation performance. Specifically, we give the definition of Precision@k and Recall@k as $Precision@k = \frac{1}{|\mathcal{U}^t|}\sum_{u_i^t \in \mathcal{U}^t} \frac{|TopK(u_i^t) \cap user(u_i^t)|}{|TopK(u_i^t)|}$ and $Recall@k = \frac{1}{|\mathcal{U}^t|}\sum_{u_i^t \in \mathcal{U}^t} \frac{|TopK(u_i^t) \cap user(u_i^t)|}{|user(u_i^t)|}$. where $TopK(u_i^t)$ is the set of friends recommended to user u_i^t on target platform that u_i^t has not yet formed links in the training set. $user(u_i^t)$ indicates the set of users that have been formed links in testing set. In our experiment, k is set to 5 and 10, respectively.

5.2.2 Performance comparison of Friend Recommendation. We compare CrossFire with several state-of-the-art friend recommendation algorithms. MF, CMF, and Tri-NMF are the personalized friend recommendation methods for **single platform**, and CFR is a **cross platform** friend recommendation method. Note that CMF is also the baseline of **joint** friend and item recommendation method.

- **RAND:** This method recommends user links randomly.
- **MF:** Matrix factorization based method which factorizes the link matrix $\mathbf{A}$ into two low rank latent matrices and predicts the links by the matrix reconstructed by them.
- **CMF:** Collective matrix factorization [25] is a matrix factorization model that jointly utilizes user-user social relation matrix $\mathbf{A}_2$ and user-item preference matrix $\mathbf{R}_2$. Note that user links are predicted as $\mathbf{U}_2^T\mathbf{O}_2$.

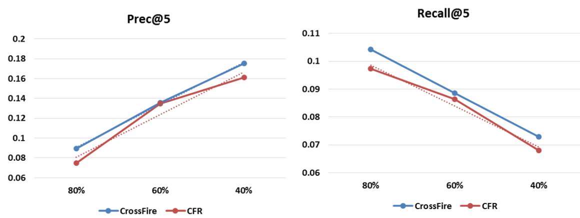
$$\min_{\mathbf{U}_2,\mathbf{V}_2,\mathbf{O}_2} \alpha\|\mathbf{Y}_2 \odot (\mathbf{A}_2 - \mathbf{U}_2^T\mathbf{O}_2)\|_F^2 + \beta\|\mathbf{W}_2 \odot (\mathbf{R}_2 - \mathbf{U}_2^T\mathbf{V}_2)\|_F^2 + \lambda(\|\mathbf{U}_2\|_F^2 + \|\mathbf{V}_2\|_F^2 + \|\mathbf{O}_2\|_F^2) \tag{21}$$

- **Tri-NMF:** Nonnegative Matrix Tri-Factorization decompose the link matrix $\mathbf{A}$ into two low rank matrices, i.e., user latent matrix $\mathbf{U}$ and user interaction matrix $\mathbf{P}$. The user links are predicted as $\mathbf{U}_2^T\mathbf{P}\mathbf{U}_2$.
- **CFR:** CFR is a variant of our proposed method without item feature sparse learning and cross site item recommendation, which has the following optimization form.

We use cross validation to determine all the model parameters. For CrossFire, we set latent dimension as $K = 10$, item sparse dimension $p = 256$. We also set $\alpha = 0.001, \beta = 1, \gamma = 0.001, \lambda = 0.01, \mu = 0.001, \nu = 1$. The experimental results are shown in Figure 2 and Figure 3. We have the following observations,

• In general, with the increase of training ratio, the recommendation performance of prec@5 and prec@10 decreases. The reason is that 1) the set of new friend relations are different for different $x\%$; 2) the difficulty of inferring new friend relations increase as the high sparsity of trust relations when training ratio is high, which can be supported by the performance of RAND. This observation is also consistent with previous work [28]. In addition, recall@5 and recall@10 increase with the increase of training ratio. The

[5] http://www.cse.msu.edu/ tangjili/trust.html



(a) Prec@5 (b) Recall@5

Figure 4: Model robustness for friend recommendation.

reason is that both truly inferred friends and remaining friends are decreasing and the latter decrease faster.

• CMF performs slightly better than MF, which shows that incorporating rating patterns to learn user latent features can help improve friend recommendation performance. In addition, Tri-NMF performs much better than MF, which indicates that factorizing user links with interaction matrix can better capture the user connection status. Moreover, cross media friend recommendation method CFR performs better than other single-platform methods.

• The proposed method CrossFire achieves the best performance comparing with other baselines. For example, CrossFire gain 54.9%, 54.9%, 46.8%,46.8% relative improvement compared with MF, in terms of $Prec@5$,$Recall@5$,$Prec@10$,$Recall@10$ respectively. It indicates that the combination of cross-media and joint prediction can provide complementary information for friend recommendations.

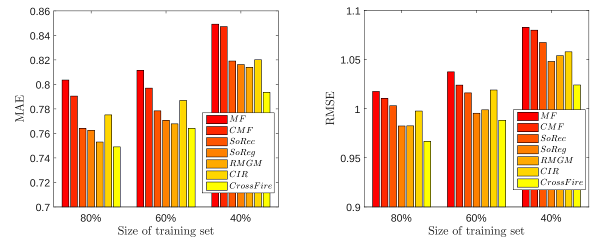
• The proposed CrossFire is more robust compared with the best baseline CFR. In Figure 4, we can see that as training ratio changes, the performance change tendencies (represented by the dotted lines) in terms of $prec@5$ and $recall@5$ are more flat. This indicates that CrossFire is less sensitive to training data size and thus can better handle data sparsity problems for friend recommendations.

5.3 Item Recommendation

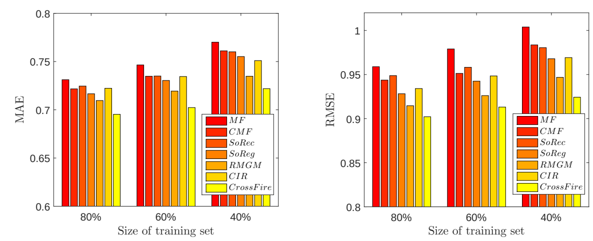
In this subsection, we further check whether the proposed framework CrossFire can improve the performance of rating predictions.

5.3.1 Experimental Settings. We randomly choose a fraction of $x\%$ positive user-item pairs for training and use the remaining $1-x\%$ of all items for testing, where x is varied in $\{80, 60, 40\}$. We use two popular metrics, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), to measure the recommendation quality of our proposed approach comparing with other collaborative filtering and social recommendation methods. The metric MAE is defined as $MAE = \frac{1}{T}\sum_{i,j}|\mathbf{R}_{ij} - \hat{\mathbf{R}_{ij}}|$, where $\mathbf{R}_{ij}$ denotes the observed rating user i gave to item j, and $\hat{\mathbf{R}_{ij}}$ denotes the predicted rating, and T is the number of tested ratings. The RMSE is defined as $RMSE = \sqrt{\frac{1}{T}\sum_{i,j}(\mathbf{R}_{ij} - \hat{\mathbf{R}_{ij}})^2}$. A smaller RMSE or MAE value means better performance. Note that previous work demonstrated that *small improvement in RMSE and MAE terms can have a significant impact on the quality of top-few recommendation* [7].

5.3.2 Performance comparison of Item Recommendation. We compare the proposed framework CrossFire with the following state-of-the-art item recommendation methods, i.e. , four **single platform** method MF, CMF, SoRec, and SoReg, and two **cross platform** transfer learning method, RMGM [10] and CIR. CMF is also the baseline of **joint** friend and item recommendation method.



(a) MAE on Ciao (b) RMSE on Ciao



(c) MAE on BookLikes (d) RMSE on BookLikes

Figure 5: MAE, RMSE performances on *Ciao* and *BookLikes*.

- **MF:** It decomposes $\mathbf{R}$ into two low rank latent matrices and predicts the ratings by the matrix reconstructed by them. It is a variant of CrossFire without considering user relations, item attributes, and cross platform similarities.
- **CMF:** Collective matrix factorization [25] is a matrix factorization model that jointly utilizes user-user social relation matrix $\mathbf{A}_2$ and user-item preference matrix $\mathbf{R}_2$. Note that item rating matrix is predicted as $\mathbf{U}_2^T\mathbf{V}_2$ as in Eqn 21.
- **SoRec:** This method [14] performs a co-factorization in user-rating matrix and user-user degree centrality relation confidence matrix by sharing same user latent factor through a probabilistic matrix factorization model.
- **SoReg:** This method [15] is based on matrix factorization model and add a social regularization term as constraints to encode local social relation of users.
- **RMGM**: Rating matrix generative model [10] is the state-of-the-art transfer learning method, which learns a shared cluster-level user ratings patterns by jointly modeling rating matrices on multiple domains.
- **CIR**: CIR is a variant of our proposed method without cross site friend recommendation.

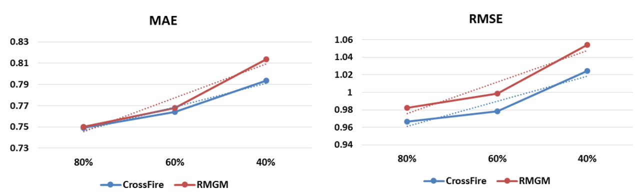
Note that we also use cross-validation to determine the parameters for all baseline methods. For CrossFire, we set the latent factor dimension as $K = 10, \alpha = 1, \beta = 0.001, \gamma = 0.001, \lambda = 0.01, \mu = 0.001, \nu = 1$. The comparison results are demonstrated in Figure 5, and we have following observations,

• Exploiting social relations as axillary information can help improve item recommendations. For example, CMF, SoRec, SoReg all performs better than MF. Note that the performance of CMF is worse than SoRec and SoReg. The reason is that SoRec and SoReg are both using social relation as side information to improve item recommendation performance; while CMF can perform joint friend and item recommendation and directly factorize user link matrix may not provide so much useful knowledge.

• Exploiting cross-media information can significantly improve recommendation performances. We can see that RMGM and CIR perform significantly better than MF in terms of MAE and RMSE in all cases. In addition, we can see that RMGM>CIR holds in all cases, which indicates that modeling cluster-level rating pattern help more than item-level rating patterns.

• The proposed method CrossFire performs the best on both datasets in terms of MAE and RMSE on all training settings. For example, CrossFire obtains 5.41% relative improvement in terms of RMSE in Ciao with 40% as the training set. The major reason is that CrossFire exploits both cross-media information and joint prediction for recommendations, which have complementary information to improve item recommendations.

• The proposed CrossFire is more robust compared with the best baseline RMGM. As shown in Figure 6, we can see that as training ratio decreases, the performance decrease tendencies (represented by the dotted lines) are more flat. This indicates that CrossFire is less sensitive to training data size and thus can better handle data sparsity problems for item recommendations.



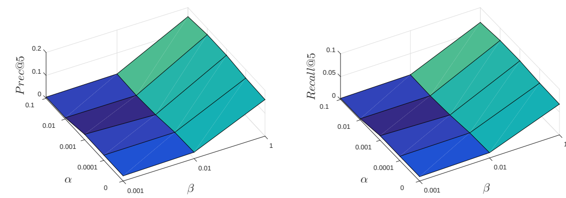
(a) MAE (b) RMSE

Figure 6: Model robustness for item recommendation.

To sum up, we conclude from the experiments that (1) the proposed framework significantly improves both friend and item recommendation performances; and (2) modeling joint prediction and cross media recommendation simultaneously provides complementary contributions to recommendation performance improvement.

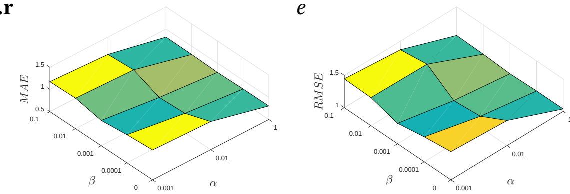
5.4 Parameter Analysis

In this section, we perform the parameter analysis for the proposed framework. We mainly focus on the parameter sensitivities for α and β, as they are controlling the joint cross friend and item recommendation components, respectively. We fix other parameters when we change the α or β. Due to the space limitation and similar observation for other settings, we only show the results when training ratio is 40% and omit the results for *prec*@10 and *recall*@10. For item recommendation, we vary the values of α as $\{1, 0.01, 0.001\}$ and β as $\{0, 0.0001, 0.001, 0.01, 0.1\}$. Similarly, for friend recommendation, we vary the values of β as $\{1, 0.01, 0.001\}$ and α as $\{0, 0.0001, 0.001, 0.01, 0.1\}$. The results for friend and item recommendations are shown in Figure 7 and Figure 8 respectively. We have the following observations: (1) When $\alpha = 1$, item recommendation has relatively good performance; however, when $\alpha = 0.01, 0.001$, the performance is much worse than MF. The reason is that $\alpha = 1$ means that the cross item recommendation part dominants the feature learning process and the resultant latent features are mainly encoded by rating information. Similarly, $\beta = 1$ ensures that latent user features are mainly encoded by user relations; (2) The performance of item recommendation is generally better when the value of β is within $[0.001, 0.01]$; similarly, for friend recommendation, α within $[0.001, 0.01]$ gives better performance. These observations ease the parameter selection process.



(a) Prec@5 (b) Recall@5

Figure 7: Parameter sensitivity on friend recommendation w.r *e*



(a) MAE (b) RMSE

Figure 8: Parameter sensitivity on item recommendation w.r.t. α and β on MAE and RMSE.

6 RELATED WORK

In this section, we introduce some related work for this paper. It mainly falls into following categories: 1) Cross Social Media Recommendations; 2) Joint friend and item recommendations.

Cross Social Media Recommendations. Cross-domain recommendations are attracting lots of attention in recent years [4]. The underly assumption of cross-domain recommendation is that there are shared properties between user and/or item profiles across source and target domains, either *explicitly* or *implicitly*. Explicit correspondences are often costly to obtain and some user identity linkage methods are proposed to bridge the gap of user links [24, 39, 40]. User correspondences can enrich user profiles and help improve recommendations in various applications, such as video recommendation [3], friend recommendation [36], product recommendation [13], etc. Implicit correspondences can be captured by modeling item similarities or even latent rating behaviors. A third-party platform (e.g. Wikipedia) can be used to bridge links of users/hashtags between Twitter and Weibo via random walk [11]. In addition, transfer learning based on collaborative filtering has recently been proposed to capture shared rating behaviors across domains, such as codebook transfer model (CBT) [9] and rating-matrix generative model (RMGM) [10]. In this paper, we aim to capture latent shared features for items and further transfer knowledge based on rating behavior patterns.

Joint friend and item recommendations. Exploiting auxiliary information to build recommendation systems are commonly used and has shown to be effective [31–34]. For example, the existing work that modeling user rating and user relation information aims to provide better recommendations for items, i.e., social recommendations [29, 30], which have been shown to be effective in recent years [14, 15, 28]. Social relations are modeled in different ways, such as trust ensemble [16], trust propagation [5], social regularization [15, 28] and matrix co-factorization [14, 28]. However, how to exploit user-item interactions for user-user link predictions has remained relatively limited [23, 37]. Even though co-factorization methods suggest that users share latent features in rating space and the social space, they can not be directly applied to user link predictions due to the indirect modeling process of social relations (e.g. social similarity matrix [15]). In this paper, we provide a collective matrix factorization [25] to directly factorize user-user and user-item matrices, which provides a novel solution to exploit the mutual benefit of rating behaviors and social relations.

7 CONCLUSION AND FUTURE WORK

Newly launched social media sites often face data sparsity challenges to perform personalized friend and item recommendations. We propose to exploit both cross-platform and within-platform information simultaneously and build a joint item and friend recommendation framework. Our framework *CrossFire* highlights three components: i) sparse coding for items, ii) matrix tri-factorization for user-user relations and user-item ratings, and iii) maintaining the factors shared across sites. Experimental results on real-world datasets demonstrate the effectiveness of proposed framework and the importance of combining cross-media recommendations and joint friend and item predictions for better recommendations.

There are several interesting future directions. First, we can consider the scenario that limited explicit user/item correspondences are given and how we can utilize this information. Second, how to perform item and friend recommendations in a streaming way for more practical use in real world cases. Third, it's also worth to explore cross-media recommendations between different domains, such as books and movies, to understand the capacity and limitations on transfer learning techniques.

8 ACKOWLEDGMENTS

This material is based upon work supported by, or in part by, the ONR grant N00014-16-1-2257.

REFERENCES

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.
[2] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.
[3] Zhengyu Deng, Jitao Sang, and Changsheng Xu. 2013. Personalized video recommendation based on cross-platform user modeling. In *ICME'13*. IEEE, 1–6.
[4] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*.
[5] Mohsen Jamali and Martin Ester. 2009. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *KDD'09*. ACM, 397–406.
[6] Alan Jennings and John J McKeown. 1992. *Matrix computation*. John Wiley & Sons Inc.
[7] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD'08*. ACM, 426–434.
[8] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. 2006. Efficient sparse coding algorithms. In *Advances in neural information processing systems*. 801–808.
[9] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.. In *IJCAI*, Vol. 9. 2052–2057.
[10] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *ICML'09*. ACM, 617–624.
[11] Xiaozhong Liu, Tian Xia, Yingying Yu, Chun Guo, and Yizhou Sun. 2016. Cross Social Media Recommendation. In *ICWSM'16*.
[12] Mingsheng Long, Guiguang Ding, Jianmin Wang, Jiaguang Sun, Yuchen Guo, and Philip S Yu. 2013. Transfer sparse coding for robust image representation. In *CVPR'13*. 407–414.
[13] Chun-Ta Lu, Sihong Xie, Weixiang Shao, Lifang He, and Philip S Yu. Item recommendation for emerging online businesses.
[14] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM'08*. ACM, 931–940.
[15] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM'11*. ACM, 287–296.
[16] Nan Ma, Ee-Peng Lim, Viet-An Nguyen, Aixin Sun, and Haifeng Liu. 2009. Trust relationship prediction using online product review data. In *Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management*. ACM, 47–54.
[17] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. *Sociological Methods & Research* 22, 1 (1993), 127–151.
[18] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[19] Neal Parikh, Stephen Boyd, and others. 2014. Proximal algorithms. *Foundations and Trends® in Optimization* 1, 3 (2014), 127–239.
[20] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
[21] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW'01*. ACM, 285–295.
[22] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR'02*. ACM, 253–260.
[23] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 3.
[24] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. 2016. User Identity Linkage across Online Social Networks: A Review. *ACM SIGKDD Explorations Newsletter* (2016).
[25] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *KDD'08*. ACM, 650–658.
[26] Parag Singla and Matthew Richardson. 2008. Yes, there is a correlation:-from social networks to personal behavior on the web. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 655–664.
[27] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.
[28] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting Local and Global Social Context for Recommendation.. In *IJCAI*. 264–269.
[29] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *Social Network Analysis and Mining* 3, 4 (2013), 1113–1133.
[30] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. 2016. Recommendation with Social Dimensions.. In *AAAI*. 251–257.
[31] Suhang Wang, Jiliang Tang, and Huan Liu. 2015. Toward dual roles of users in recommender systems. In *CIKM*. ACM, 1651–1660.
[32] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. 2015. Exploring Implicit Hierarchical Structures for Recommender Systems.. In *IJCAI*. 1813–1819.
[33] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *WWW*. International World Wide Web Conferences Steering Committee, 391–400.
[34] Yilin Wang, Suhang Wang, Jiliang Tang, Guo-Jun Qi, Huan Liu, and Baoxin Li. 2017. CLARE: A Joint Approach to Label Classification and Tag Recommendation.. In *AAAI*. 210–216.
[35] Zaiwen Wen and Wotao Yin. 2013. A feasible method for optimization with orthogonality constraints. *Mathematical Programming* 142, 1-2 (2013), 397–434.
[36] Ming Yan, Jitao Sang, Tao Mei, and Changsheng Xu. 2013. Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge. In *ICME'13*. IEEE, 1–6.
[37] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. 2011. Like like alike: joint friendship and interest propagation in social networks. In *WWW'11*. ACM, 537–546.
[38] Dani Yogatama and Noah A Smith. 2014. Making the Most of Bag of Words: Sentence Regularization with Alternating Direction Method of Multipliers.. In *ICML'14*. 656–664.
[39] Reza Zafarani and Huan Liu. 2009. Connecting Corresponding Identities across Communities. *ICWSM* 9 (2009), 354–357.
[40] Reza Zafarani and Huan Liu. 2013. Connecting users across social media sites: a behavioral-modeling approach. In *KDD'13*. ACM, 41–49.
[41] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. 2011. Graph regularized sparse coding for image representation. *IEEE Transactions on Image Processing* 20, 5 (2011), 1327–1336.