

# Personalized Entity Recommendation: A Heterogeneous Information Network Approach

Xiao Yu<sup>†</sup>, Xiang Ren<sup>†</sup>, Yizhou Sun<sup>‡</sup>, Quanquan Gu<sup>†</sup>,  
Bradley Sturt<sup>†</sup>, Urvashi Khandelwal<sup>†</sup>, Brandon Norick<sup>†</sup>, Jiawei Han<sup>†</sup>

<sup>†</sup>University of Illinois at Urbana-Champaign, Urbana, IL, USA

<sup>†</sup>{xiaoyu1, xren7, qgu3, sturt1, khndlw12, bnorick, hanj}@illinois.edu

<sup>‡</sup>Northeastern University, Boston, MA, USA

<sup>‡</sup>yzsun@ccs.neu.edu

## ABSTRACT

Among different hybrid recommendation techniques, network-based entity recommendation methods, which utilize user or item relationship information, are beginning to attract increasing attention recently. Most of the previous studies in this category only consider a single relationship type, such as friendships in a social network. In many scenarios, the entity recommendation problem exists in a *heterogeneous information network* environment. Different types of relationships can be potentially used to improve the recommendation quality. In this paper, we study the entity recommendation problem in heterogeneous information networks. Specifically, we propose to combine heterogeneous relationship information for each user differently and aim to provide high-quality personalized recommendation results using user implicit feedback data and personalized recommendation models.

In order to take full advantage of the relationship heterogeneity in information networks, we first introduce meta-path-based latent features to represent the connectivity between users and items along different types of paths. We then define recommendation models at both global and personalized levels and use Bayesian ranking optimization techniques to estimate the proposed models. Empirical studies show that our approaches outperform several widely employed or the state-of-the-art entity recommendation techniques.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

## Keywords

Hybrid Recommender System; Information Network

## 1. INTRODUCTION

Entity recommendation, as one of the most effective and widely used information filtering and discovery methods, has been actively studied in the past decade in both industry and

academia. Among different recommendation techniques, hybrid recommender systems, which combine user feedback data (explicit ratings or implicit user log) and additional information of users or items, can achieve better recommendation results in certain scenarios, based on recent studies [6] [20].

In many recommendation applications, **additional information regarding users and items can be obtained**, e.g., user demographic attributes, product specifications, or user social network information. Moreover, the entity recommendation problem often exists in a heterogeneous information network environment with different types of attributes and relationships of users, items, and other entities available. An illustration of a movie recommendation problem in a **heterogeneous information network** can be found in Figure 1. In this example, besides users and movies and the user-movie interaction relationships, other types of entities can be found and linked to the movie recommendation problem, such as actors, directors, and genres. Different types of relationships between users (e.g., friendship) and entities (e.g., director directed a movie) can be potentially utilized in hybrid recommender systems.

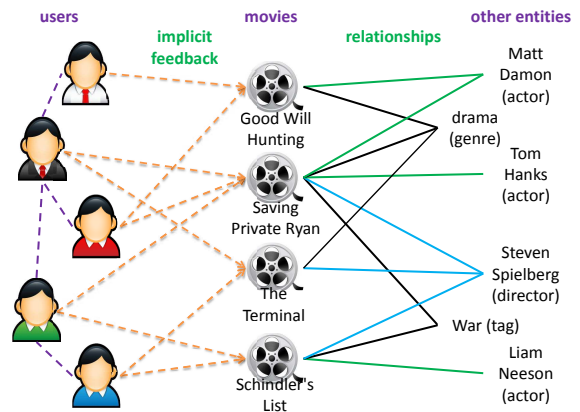


Figure 1: A heterogeneous information network snippet with users, movies, actors, directors and tags as entities and the corresponding relationships between these entities

Previous studies suggest that by utilizing additional user or item relationship information, the quality of the recommender systems can be improved. Our study falls in the category of such hybrid recommender systems. The difference between our work and other link-based hybrid methods is that most previous studies only utilize a single type of relationship, e.g., trust relationship [10], friend relationship [19],

or user membership [31]. We propose to study entity recommendation problem in the aforementioned heterogeneous network environment, **aiming to take advantage of different types of relationship information at the same time.**

Previous studies [29] [30] on link-based hybrid recommender systems apply the same recommendation models to all the users when recommending items. They rely on the personal ratings or user feedback data to achieve recommendation personalization. However, such approaches cannot fully distinguish user interests and preferences, and thus may lead to unsatisfying results. For example, Alice and Bob watched the movie “Pacific Rim”. Alice watched this movie because she likes robot / monster stories (genre and story of the movie) while Bob watched this movie because his friends watched the same movie. If we apply the same recommendation model (e.g., social network-based collaborative filtering) to both users without understanding or differentiating their motives and interests, the recommendation results may not satisfy the information discovery needs of different users.

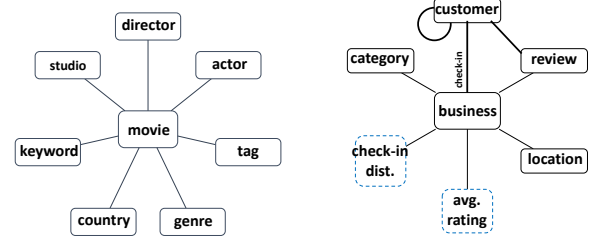
In this paper, we introduce a novel entity recommendation framework in heterogeneous information networks using implicit feedback data. **We combine user feedback with different types of entity relationships in a collaborative filtering way.** We personalize recommendation results by both considering personal user implicit feedback data and building personalized recommendation models for different users.

To take advantage of the relationship heterogeneity of the information network, we first diffuse the observed user implicit feedback along different meta-paths to generate possible recommendation candidates, under the corresponding user interest semantic assumptions. We apply matrix factorization techniques on the diffused user preferences to calculate latent representations for users and items accordingly. We then combine these latent features and define a global recommendation model. To further distinguish user interests, we propose to build recommendation models at personalized level, i.e., **we build different entity recommendation models for different users.** We adopt a Bayesian ranking optimization technique for model estimation. Empirical studies in two real world datasets, IMDb-MovieLens-100K and Yelp, have shown that the proposed recommendation models outperform several state-of-the-art implicit feedback recommendation systems.

The contributions of this paper are summarized as follows:

1. We study personalized entity recommendation with implicit user feedback in heterogeneous information networks.
2. To take advantage of the relationship heterogeneity, we propose to diffuse user preferences along different meta-paths in information networks to generate latent features for users and items.
3. The proposed framework generates personalized recommendation models for different users effectively and efficiently.
4. Empirical studies in two real-world datasets, IMDb-MovieLens-100K and Yelp, demonstrate the power of our methodology.

The remainder of this paper is organized as follows. The preliminaries for this study are introduced in Section 2. The meta-path-based latent features and the global recommendation model are presented in Section 3. We propose the personalized recommendation framework in Section 4 and parameter estimation techniques in 5. Experiments and analysis are in Section 6. Finally, we discuss related work in Section 7 and conclude the study in Section 8.



(a) IMDb Network Schema (b) Yelp Network Schema

**Figure 2: Information network schemas (solid rectangles represent entity types and dashed rectangles represent attribute types)**

**Table 1: Notations**

Notation	Description
$u, e$	user, item (or entity)
$R$	implicit feedback matrix
$G, G_T$	heterogeneous information network and schema
$A, R$	entity type and relationship type
$\mathcal{P}, p$	meta-path and path
$R^{(q)}$	diffused user preferences along the $q$ -th meta-path
$U, V$	low rank representations of users and items
$C$	user clusters
$\theta, \theta^{(i)}$	global and hidden local model parameters

## 2. BACKGROUND AND PRELIMINARIES

In this section, we present the background and preliminaries of this study. Detailed problem definition is included at the end of this section.

### 2.1 Binary User Feedback

With  $m$  users  $\mathcal{U} = \{u_1, \dots, u_m\}$  and  $n$  items  $\mathcal{I} = \{e_1, \dots, e_n\}$ , we define the user implicit feedback matrix  $R \in \mathbb{R}^{m \times n}$  as follows:

$$R_{ij} = \begin{cases} 1, & \text{if } (u_i, e_j) \text{ interaction is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

Notice that the value 1 in  $R$  represents interactions between users and items, e.g., users watched a movie or users browsed a restaurant website. **The value 1 in the implicit feedback data does not mean that users like the items.** A user buys a movie ticket because she or he is interested in the movie but this user might dislike the movie after watching it. **Similarly the value 0 in  $R$  does not mean that the users dislike the items,** but are a mixture of negative feedback (the users are not interested in the items) and unobserved interactions (the users are not aware of these items for now). Several previous studies have additional assumptions about the implicit feedback data, e.g., user-item interaction frequency, or the dwelling time of each interaction. Not to digress from the purpose of this study, we use binary user feedback in its original form as defined above. However, additional information as mentioned can be added into the factorization process of the proposed models accordingly.

### 2.2 Heterogeneous Information Network

Similar to [24], we define information networks as follows:

**DEFINITION 1 (INFORMATION NETWORK).** *An information network is defined as a directed graph  $G = (V, E)$  with an entity type mapping function  $\phi : V \rightarrow \mathcal{A}$  and a link type mapping function  $\psi : E \rightarrow \mathcal{R}$ . Each entity  $v \in V$  belongs to*

meta path

$P_1: u \xrightarrow{\text{Viewed}} \text{Movie} \xrightarrow{\text{Viewed}^{-1}} u \xrightarrow{\text{follows}} \text{Actor} \xrightarrow{\text{starred}} \text{Movie}$

an entity type  $\phi(v) \in \mathcal{A}$ , and each link  $l \in E$  belongs to a relation type  $\psi(l) \in \mathcal{R}$ .

We call an information network a *heterogeneous information network* when  $|\mathcal{A}| > 1$  or  $|\mathcal{R}| > 1$ . To be consistent with recommender system terminology, we refer to the entities in the information network which are being recommended as *items*.

Similar to an entity-relation diagram in a relational database, we use an abstract graph (i.e., *network schema*) to represent the entity and relation type restrictions in heterogeneous information networks, denoted by  $G_T = (\mathcal{A}, \mathcal{R})$ . Examples of heterogeneous information networks and partial network schemas can be found in Figure 2.

### 2.3 Matrix Factorization for Implicit Feedback

Matrix factorization techniques have been used to interpret implicit user feedback in previous studies [5] [9], by learning the low-rank matrix representations for users and items. More specifically, factorization methods seek to approximate the implicit feedback matrix  $R$  with the product of the low-rank matrices as follows:

$$R \approx UV^T \quad (1)$$

where  $U \in \mathbb{R}^{m \times d}$  are the latent features representing users and  $V \in \mathbb{R}^{n \times d}$  are the latent features representing items, with  $d < \min(n, m)$ .

The recommendation score between  $u_i$  and  $e_j$  can be computed with the estimated low-rank matrices as  $r(u_i, e_j) = U_i V_j^T$ , where  $U_i$  denotes the  $i$ -th row of the matrix  $U$  and  $V_j$  denotes the  $j$ -th row of  $V$ . By sorting items with their recommendation scores, we can return the top- $k$  items which  $u_i$  has not interacted with before as the recommendation results.

To solve Equation (1), the non-negative factorization techniques (NMF) discussed in [5] can be directly employed. More advanced methods [9] [6] [19] have been studied recently to incorporate additional information in order to further improve the performance.

In our proposed recommendation models, when defining user item features, we rely on matrix factorization methods to derive low-rank representations of users and items under different semantics. One should notice that our proposed models are orthogonal to factorization techniques, i.e., one can extend the proposed models easily with advanced factorization techniques. In this study, aiming to propose a generic recommendation framework, we use the basic NMF method in [5] when defining features and models. By utilizing advanced factorization methods, the performance of the our methods can be improved accordingly due to the aforementioned orthogonality.

### 2.4 Problem Definition

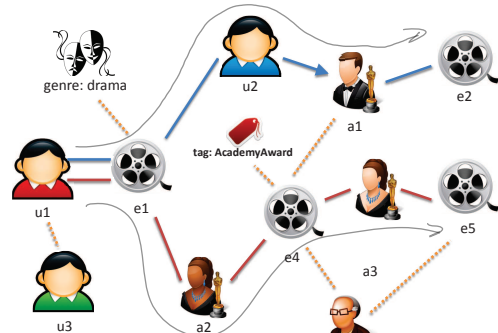
We define the recommendation problem which we study in this paper as follows:

**DEFINITION 2 (PROBLEM DEFINITION).** Given a heterogeneous information network  $G$  with user implicit feedback  $R$ , for a user  $u_i$ , we aim to build personalized recommendation model for  $u_i$ , and recommend a ranked list of items that are of interest to  $u_i$  accordingly.

Notations which are used in the rest of the paper can be found in Table 1.

## 3. META-PATH-BASED LATENT FEATURES

In this section, aiming to utilize the rich yet under-discovered information network, we present a user preference diffusion-based feature generation approach, which combines user implicit feedback and heterogeneous entity relationships. We



$P_2: u \xrightarrow{\text{Viewed}} \text{Movie} \xrightarrow{\text{starred}^{-1}} \text{Actor} \xrightarrow{\text{starred}} \text{Movie}$

**Figure 3: User preference discussion along different meta-paths (Example 1)**

then define a recommendation function with the latent features at a global level at the end of this section. Notice that we use the term “global” to represent the process of applying the same recommendation model to all users. We still utilize personal user implicit feedback data during recommendation with the global recommendation model. We introduce personalized recommendation models in Section 4. We discuss the learning algorithms for the recommendation models in Section 5.

### 3.1 Meta-Path

From the information network point of view, the entity recommendation problem is to seek certain connectivity between users and items. In heterogeneous information networks, two entities can be connected via different paths (see Figure 3 for examples). Due to the multiplicity of entity and relationship types in the information networks, these paths may contain different entity types, relationship types in different orders and they can have various lengths. In order to describe path types in heterogeneous information network, we introduce the definition of *meta-path* proposed in [24]. Meta-paths are defined in the scope of information network schemas, and describe how two entity types could be connected via different types of paths.

**DEFINITION 3 (META-PATH).** A meta path  $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \dots \xrightarrow{R_k} A_k$  is a path in a network schema  $G_T = (\mathcal{A}, \mathcal{R})$ , which defines a new composite relation  $R_1 R_2 \dots R_k$  between type  $A_0$  and  $A_k$ , where  $A_i \in \mathcal{A}$  and  $R_i \in \mathcal{R}$  for  $i = 0, \dots, k$ ,  $A_0 = \text{dom}(R_1) = \text{dom}(\mathcal{P})$ ,  $A_k = \text{range}(R_k) = \text{range}(\mathcal{P})$  and  $A_i = \text{range}(R_i) = \text{dom}(R_{i+1})$  for  $i = 1, \dots, k-1$ .

where  $\text{dom}(\cdot)$  defines the domain of certain relationship and  $\text{range}(\cdot)$  defines the range.

We use  $p$  to denote the paths in information networks and  $\mathcal{P}$  to denote meta-paths. Based on the above definition, one can notice that meta-paths are the types for paths in information networks. Previous studies suggest that meta-paths can be used to facilitate entity similarity or proximity measurement and similarity semantic disambiguation [24] [28]. We use the following example to demonstrate the intuition of utilizing meta-paths in entity recommendation problem.

**EXAMPLE 1 (DIFFERENT META-PATHS IN IMDB).** With the graph schema of IMDB defined in Figure 2(a), we can derive a number of meta-paths which connect users with movies. We show two possible meta-paths as follows:



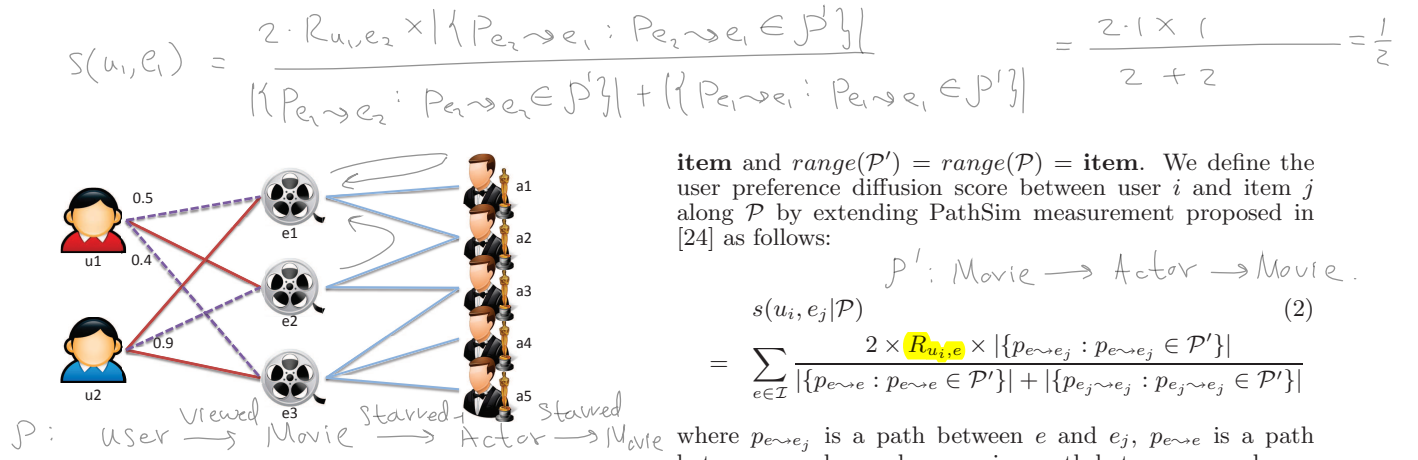


Figure 4: User preference diffusion score calculation (Example 2). The solid red links represent observed user implicit feedback while the purple dotted links represent diffused user preferences.

- $\mathcal{P}_1$ : user  $\xrightarrow{\text{Viewed}}$  movie  $\xrightarrow{\text{Viewed}^{-1}}$  user  $\xrightarrow{\text{Follows}}$  actor  $\xrightarrow{\text{StarredIn}}$  movie
- $\mathcal{P}_2$ : user  $\xrightarrow{\text{Viewed}}$  movie  $\xrightarrow{\text{StarredIn}^{-1}}$  actor  $\xrightarrow{\text{StarredIn}}$  movie  $\xrightarrow{\text{StarredIn}^{-1}}$  actor  $\xrightarrow{\text{StarredIn}}$  movie

We give two path examples of the two meta-paths in Figure 3 (the solid blue line represents meta-path  $\mathcal{P}_1$  and the solid red line represents meta-path  $\mathcal{P}_2$ ). These two meta-paths connect users and movies under different semantic assumptions.  $\mathcal{P}_1$  explores the social relationships of other users who watched movies in common with the target user, while  $\mathcal{P}_2$  utilizes movie-actor links to build relationships between users and movies. By measuring proximity between users and movies along different meta-paths, we may be able to make movie recommendations to users from different semantic perspectives in information networks.

When representing longer meta-paths, relationship types can be ignored when doing so does not cause ambiguity. The recursive parts of the meta-paths can be compressed with exponentiation notation. For example, we use **user** – (**movie** – **actor** – **movie**)<sup>2</sup> to represent meta-path  $\mathcal{P}_2$  in the above example.

### 3.2 User Preference Diffusion

With the implicit user feedback data defined in Section 2 and meta-path defined above, we introduce the user preference diffusion process along meta-paths. As mentioned before, implicit feedback represents user item interactions. the value 1 in implicit feedback indicates that users are more interested in the corresponding items than the rest of the items. We use the term *user preference* to represent the user interests in implicit feedback data. Intuitively, if we can understand the semantic meanings of user preferences and find similar items to the ones that the users were interested in, following the discovered semantics, we can make entity recommendations to these users accordingly.

Based on this observation and the problem definition presented in Section 2, in this paper, we focus on meta-paths in the format of **user** – **item** – \* – **item** when building recommendation models. The intuition is we want to diffuse the observed users preferences in implicit feedback data along different meta-paths so that users can be connected with other items. By defining a user preference diffusion score between the target user and all possible items along different meta-paths, we can now measure the possibility of an unobserved user-item interaction in the information network under different semantic assumptions.

Given a meta-path  $\mathcal{P} = R_1 R_2 \dots R_k$  with  $\text{dom}(\mathcal{P}) = \text{user}$  and  $\text{range}(\mathcal{P}) = \text{item}$ , let  $\mathcal{P}' = R_2 \dots R_k$  with  $\text{dom}(\mathcal{P}') =$

item and  $\text{range}(\mathcal{P}') = \text{range}(\mathcal{P}) = \text{item}$ . We define the user preference diffusion score between user  $u_i$  and item  $j$  along  $\mathcal{P}$  by extending PathSim measurement proposed in [24] as follows:

$$s(u_i, e_j | \mathcal{P}) = \frac{2 \times R_{u_i, e} \times |\{p_{e \rightarrow e_j} : p_{e \rightarrow e_j} \in \mathcal{P}'\}|}{|\{p_{e \rightarrow e} : p_{e \rightarrow e} \in \mathcal{P}'\}| + |\{p_{e_j \rightarrow e_j} : p_{e_j \rightarrow e_j} \in \mathcal{P}'\}|} \quad (2)$$

where  $p_{e \rightarrow e_j}$  is a path between  $e$  and  $e_j$ ,  $p_{e \rightarrow e}$  is a path between  $e$  and  $e$ , and  $p_{e_j \rightarrow e_j}$  is a path between  $e_j$  and  $e_j$ .

The user preference diffusion score between user  $u_i$  and item  $e_j$  contains two parts: (1) the observed user-item interactions associated with  $u_i$ , and (2) the connectivity between the items that  $u_i$  is interested in and potential items of interest, which are represented by  $e_j$  in Equation (2). Notice the connectivity between items is defined as the number of paths between these items following meta-path  $\mathcal{P}$  and normalized by the visibility of the items so the diffusion score does not overly favor popular items. We demonstrate the user preference diffusion process with a toy example as in Figure 4.

#### EXAMPLE 2 (USER PREFERENCE DIFFUSION SCORE).

In this toy example, we use a small information network which contains two users ( $u_1$  and  $u_2$ ), three movies ( $e_1$ ,  $e_2$  and  $e_3$ ), and 5 actors ( $a_1, \dots, a_5$ ). These entities are interconnected as shown in Figure 4. (The solid red links represent observed user implicit feedback while the purple dotted links represent the diffused user preferences). We use **user** – **movie** – **actor** – **movie** as meta-path  $\mathcal{P}$  when calculating the diffusion score. Based on implicit feedback data  $R$ , we know that  $u_1$  watched movie  $e_2$ . Based on the information network structure, there is 1 path between  $e_1$  and  $e_2$  following the aforementioned meta-path, 2 paths between  $e_1$  and  $e_1$  and 2 paths between  $e_2$  and  $e_2$ . By plugging the implicit feedback data and the numbers of paths described above into Equation (2), we can get that the user preference diffusion score under meta-path  $\mathcal{P}$  from  $u_1$  to  $e_1$  is 0.5. Other diffusion scores in this example can be calculated accordingly.

By measuring the diffusion scores between all users and all items along meta-path  $\mathcal{P}$ , we can generate a diffused user preference matrix  $\tilde{R} \in \mathbb{R}^{m \times n}$ .  $\tilde{R}_i$  represents the possible preferences of user  $u_i$  if he or she explores the network for new content following meta-path  $\mathcal{P}$  ( $u_i$  watches movies with certain genres or  $u_i$  watches movies from certain directors). By repeating this process, with  $L$  different meta-paths, we can calculate  $L$  different diffused user preference matrices accordingly. We denote these user preference matrices as  $\tilde{R}^{(1)}, \tilde{R}^{(2)}, \dots, \tilde{R}^{(L)}$ .

This process propagates user preferences along different meta-paths in the heterogeneous information networks, and it mimics users' information discovery process. The diffusion score indicates the possibility of certain user-item interaction under certain meta-path semantic. The diffusion scores can be used to define recommendation models.

### 3.3 Global Recommendation Model

We denote  $\tilde{R}^{(q)}$  as the diffused user preference matrix along the  $q$ -th meta-path. Following the intuition and principle of matrix factorization-based recommendation methods, we can derive low-rank user and item matrices from each diffused preference matrix accordingly. These low-rank matrices are the latent representations of users and items, under the semantic meaning of the corresponding meta-path. With low-rank matrix factorization technique, we can fac-

torize the diffused matrix  $\tilde{R}^{(q)}$  as follows:

$$\begin{aligned} (\hat{U}^{(q)}, \hat{V}^{(q)}) &= \operatorname{argmin}_{U, V} \|\tilde{R}^{(q)} - UV^T\|_F^2 \\ \text{s.t.} \quad &U \geq 0, V \geq 0, \end{aligned} \quad (3)$$

where  $\hat{U}^{(q)} \in \mathbb{R}^{m \times d}$  represents users and  $\hat{V}^{(q)} \in \mathbb{R}^{n \times d}$  represents items, with  $d < \min(n, m)$ .  $\hat{U}_i^{(q)}$  is the latent feature for user  $u_i$  along the  $q$ -th meta-path and  $\hat{V}_j^{(q)}$  is the latent feature for  $e_j$  along the  $q$ -th meta-path respectively.

As discussed in Section 2, we apply the generic NMF method to solve Equation (3), although more advanced factorization techniques can be used instead. By repeating the above process for all  $L$  diffused user preference matrices, we can now generate  $L$  pairs of representations of users and items  $(\hat{U}^{(1)}, \hat{V}^{(1)}), \dots, (\hat{U}^{(L)}, \hat{V}^{(L)})$ . Each low-rank feature pair represents users and items under a specific relationship semantic due to the user preference diffusion process. When defining recommendation models with these latent-features, different feature pairs may have different importance. For example, users are more likely to follow certain actors when choosing movies rather than consider which movie studios these movies are made. With this observation, following [30], we define a global recommendation model as follows:

$$r(u_i, e_j) = \sum_{q=1}^L \theta_q \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \quad (4)$$

where  $\theta_q$  is the weight for the  $q$ -th user and item low-rank representation pair. Based on the non-negative property of the features, we add  $\theta_q \geq 0$  as an optimization constraint.

With the recommendation model in Equation (4), given a user, we can now assign recommendation scores to all items, and then rank these items accordingly. We return the top- $K$  results as the recommendation results. We will discuss how to estimate the parameters in the recommendation model in Section 5.

## 4. PERSONALIZED RECOMMENDATION MODEL

We proposed to diffuse user preferences along different meta-paths in heterogeneous information networks, and calculate latent features for users and items under various semantic assumptions with matrix factorization techniques. We then defined the global recommendation model with these latent features, which essentially combines the observed implicit feedback from users and different types of relationships in the information network together. During recommendation, we apply the global model to all users and utilize users' personal feedback to generate recommendation results. However, such a solution does not distinguish user interests or behavior patterns at the model level. For example, the learned global model may suggest that majority users watch popular movies featuring famous actors, but this rule might not be true for all individuals.

In this section, we extend the proposed global recommendation model to a finer level of granularity. Rather than learn one recommendation model for all the users, we aim to calculate different recommendation models for different users to better capture user preferences and interests. The straightforward way of learning personalized models is to estimate the recommendation model defined in Equation (4) with each user's own implicit feedback data. However, the number of feedback per user follows power law distribution (we demonstrate this claim in Section 6), which means we do not have enough feedback data to learn personalized models for most of the users.

Although users may have different behaviors from each other, a subgroup of users can share similar interests and

preferences from certain perspectives. For example, comic fans are interested in super-hero, fantasy and adventure movies while fans of Steven Spielberg follow movies directed by him. Motivated by this observation, we propose to first cluster users based on their interests, and then learn a recommendation models within each cluster. Notice that one user can belong to different user clusters (one can be comic fan and Spielberg fan at the same time). When recommending, we first calculate the personalized recommendation model for the target user by combining the recommendation models of the related user clusters, and then calculate the recommendation results with the personalized model of the target user. We define the personalized recommendation function for user  $u_i$  as follows:

$$r^*(u_i, e_j) = \sum_{k=1}^c \operatorname{sim}(C_k, u_i) \sum_{q=1}^L \theta_q^{\{k\}} \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \quad (5)$$

where  $C$  represents user clusters related to target user  $u_i$  and function  $\operatorname{sim}(\cdot, \cdot)$  defines the cosine similarity between the center of cluster  $C_k$  and  $u_i$ .  $\theta_q^{\{k\}}$  represents the recommendation model defined in cluster  $C_k$ . We define  $\theta^{\{k\}} = \theta^{\{1\}}, \dots, \theta^{\{c\}}$  as the recommendation model parameters for this approach.

Compared with  $L$  parameters in the global recommendation model (Equation (4)), the personalized recommendation approach has a total  $c \times L$  parameters, where  $c$  is the number of clusters. With a relatively larger parameter space, now we can generate personalized recommendation models efficiently and represent different user interests or behavior patterns effectively. We discuss the user clustering and the model learning algorithm in details in Section 5.

After estimating recommendation models for all the user clusters  $\theta^{\{k\}}$ , when recommending items for  $u_i$ , we first find the clusters that  $u_i$  is related to (has a high similarity to the cluster center), and then combine the related user clustering parameters following Equation (5). With the calculated personalized recommendation model, we can assign recommendation score to each item for  $u_i$ . Recommendation can be made by sorting all items with their recommendation scores and returning the top- $k$  items.

The number of clusters could be essential to this method. If the cluster number  $c$  is too small, we may not be able to distinguish user interests well. If the cluster number is too big, the number of users in each cluster becomes very small. In this case, we may not have enough training data to learn the recommendation models. A good estimation of the optimal number of clusters can be achieved by cross-validation using training data. We discuss the performance change with different parameter  $c$  in Section 6.5.

## 5. MODEL LEARNING WITH IMPLICIT FEEDBACK

In this section, we introduce learning algorithms for both global and personalized recommendation models. We first discuss parameter estimation method for global recommendation model (Equation (4)) and then extend the learning algorithm to personalized recommendation models.

Recommendation models proposed in this paper take advantage of the heterogeneous entity relationships in information networks. More specifically, we combine network diffusion-based latent features with parameters indicating the importance of the corresponding meta-path in the recommendation process. To learn the importance of the latent features, we use user implicit feedback as training data. As discussed in Section 2, the value 1 in implicit feedback data represents positive feedback (users are interested in such

items) while the value 0 represents a mixture of negative feedback (users are not interested in such items) and unobserved potential interactions (users are not aware of such items). Traditional learning methods adopt classification or learning-to-rank objective functions and usually treat 1s in training dataset as positives and 0s as negatives. As we discussed, such methods do not fit in the definition of implicit feedback data and they cannot generate high quality recommendation models.

Motivated by [21], we employ a different learning approach by considering the correct item pair orders. We define an objective function to order 1 values before 0 values for each user. The assumption behind this objective function is that users are more interested in the items with value 1 in  $R$  than the rest of the items, which is a weaker and more plausible assumption compared with the traditional approaches.

## 5.1 Bayesian Ranking-Based Optimization

We use  $p(e_a > e_b; u_i | \theta)$  to denote the probability that user  $u_i$  prefers  $e_a$  over  $e_b$ . The Bayesian formulation of the optimization criterion is to maximize the posterior probability as follows:

$$p(\theta | R) \propto p(R | \theta) p(\theta), \quad (6)$$

where  $\theta = \{\theta_1, \dots, \theta_L\}$  represents the global model parameters, and  $p(R | \theta)$  represent the probability that all item pairs can be ranked correctly for all users according to  $R$ , i.e., for each user, items with feedback 1 can be ranked before items with feedback 0.

With the assumption that both user preferences and ordering of the item pairs are independent, we can expand the likelihood function  $p(R | \theta)$  as follows:

$$\begin{aligned} p(R | \theta) &= \prod_{u_i \in \mathcal{U}} p(R_i | \theta) \\ &= \prod_{u_i \in \mathcal{U}} \prod_{(e_a > e_b) \in R_i} p(e_a > e_b; u_i | \theta) \end{aligned} \quad (7)$$

where  $(e_a > e_b) \in R_i$  represent all item pairs with the correct orders in the observed implicit feedback of  $u_i$ .

We define  $p(e_a > e_b; u_i | \theta)$  as:

$$p(e_a > e_b; u_i | \theta) = \sigma(r(u_i, e_a) - r(u_i, e_b)), \quad (8)$$

where  $\sigma$  is the logistic sigmoid function  $\sigma(x) = \frac{1}{1+e^{-x}}$ .

We assume that  $p(\theta)$  is a Gaussian distribution with zero mean and variance-covariance matrix  $\Sigma_\theta = \lambda I$ . With the probability and the likelihood defined above, we can derive the objective function as follows:

$$\begin{aligned} O &= -\ln p(\theta | R) = -\ln p(R | \theta) p(\theta) \\ &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \ln p(e_a > e_b; u_i | \theta) + \lambda \|\theta\|_2^2 \\ &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \ln \sigma(r(u_i, e_a) - r(u_i, e_b)) + \lambda \|\theta\|_2^2 \end{aligned} \quad (9)$$

where  $\lambda \|\theta\|_2^2$  is a data dependent  $L_2$  regularization term.

By minimizing  $O$  in Equation (9), we can estimate the recommendation parameter  $\theta$  from the implicit feedback data.

## 5.2 The Optimization Algorithm

Notice that Equation (9) is differentiable, many optimization techniques (e.g., SGD [2], BFGS-B method [3]) can be utilized to estimate parameter  $\theta$ . The gradient of Equa-

tion (9) with respect to  $\theta$  can be calculated as follows:

$$\begin{aligned} \frac{\partial O}{\partial \theta} &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \frac{\partial}{\partial \theta} \sigma(r_{i,ab}) + \frac{\lambda}{2} \frac{\partial}{\partial \theta} \|\theta\|_2^2 \\ &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \frac{e^{-r_{i,ab}}}{1 + e^{-r_{i,ab}}} \frac{\partial}{\partial \theta} r_{i,ab} + \lambda \theta, \end{aligned}$$

where  $r_{i,ab} = r(u_i, e_a) - r(u_i, e_b)$ .

Considering the data size of the real-world recommender systems, with the above gradient, we employed the stochastic gradient descent (SGD) method [2] to estimate the parameters in our empirical studies. Notice that the time complexity of this proposed learning process is  $O(mn^2)$  where  $m$  is the number of users and  $n$  is the number of items. In large datasets this can be overwhelming. With SGD, we only need to estimate the gradient with a very small subset ( $10^{-5}$  sample rate) of training pairs sampled from  $R$  at each iteration. We discuss sample rate selection for parameter estimation in Section 6.

## 5.3 Learning Personalized Recommendation Models

As discussed in Section 4, the proposed global recommendation model failed to distinguish individual interests and behavior differences, and thus the quality of the results may not be satisfying. We observed that although users may have different interests from each other, a subgroup of users can share similar interests or behavior patterns. Instead of learning one global model, which cannot represent user individuality, or learning personalized models directly, which can be time consuming and may cause model over-fitting, we propose to learn recommendation models for user subgroups. With recommendation models for user subgroups, when making recommendations, we can calculate the personalized recommendation model for a target user by combining the recommendation models he or she is most related to, and generate recommendation results accordingly.

In order to learn recommendation models for user subgroups, we first need to cluster users based on their interests and preferences, by examining the user implicit feedback data. Considering the sparsity of  $R$ , we first learn low-rank representation for users by applying non-negative matrix factorization on  $R$  directly. With the low dimension user matrix  $U$ , we apply the well-studied  $k$ -means algorithm with a cosine function as similarity measurement between users, to finally cluster users into subgroups. For each cluster, we apply the techniques we discussed above to learn a recommendation model. The learning algorithm of personalized recommendation models can be found in Algorithm 1.

After estimating parameters of the recommendation models, given a target user, we can calculate the corresponding personalized recommendation model with Equation (5), and make personalized entity recommendation by using both the personalized recommendation model and his or her personal feedback data accordingly.

## 6. EMPIRICAL STUDY

We present the empirical studies of the proposed recommendation framework in this section. We implemented both global and personalized recommendation models proposed in Section 3 and 4 along with several popularly deployed or the state-of-the-art implicit feedback recommendation techniques. We applied these methods on two real-world datasets and performed a series of experiments to demonstrate the effectiveness of the proposed approach. We present both experimental results and discussion with analysis.



---

**Algorithm 1: Learning Personalized Recommendation Models**


---

```

// input: implicit feedback and information
// network
// output: recommendation models for user
// clusters
Input:  $R, G$ 
Output:  $\theta^{\{ \cdot \}}$ 
Prepare  $L$  meta-paths in the format of user – item
– * – item
// User preference diffusion along meta-paths
for  $q \leftarrow 1$  to  $L$  do
    foreach  $u_i$  and  $e_j$  do
         $\tilde{R}_{u_i, e_j}^{(q)} = s(u_i, e_j | \mathcal{P}^{(q)})$  (Equation (2))
    end
    Calculate latent features  $\hat{U}^{(q)}, \hat{V}^{(q)}$  with  $\tilde{R}^{(q)}$ 
    (Equation (3))
end
// Clustering users into subgroups
Factorize  $R$  and derive  $U, V$ 
 $C = \text{k-means}(U)$ 
// Learn recommendation models
foreach  $C_k$  in  $C$  do
    Optimize  $\theta^{\{k\}}$  with implicit feedback in user
    subgroup  $C_k$  (Equation (9))
end

```

---

Name	#Items	#Users	#Ratings	#Entities	#Links
IM100K	943	1360	89,626	60,905	146,013
Yelp	11,537	43,873	229,907	285,317	570,634

(a) Datasets Description

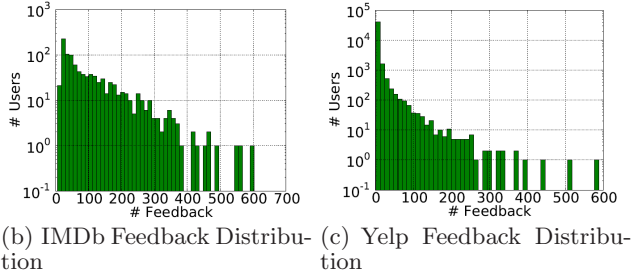


Figure 5: IM100K and Yelp Datasets

## 6.1 Data

To demonstrate the effectiveness of the proposed recommendation framework, we choose two datasets from different domains (movie and local business) for empirical studies. The first dataset is built by combining the popular MovieLens-100K dataset and the corresponding IMDb dataset together. We name this dataset IMDb-MovieLens-100K (IM100K). We use MovieLens user ratings as user feedback and we build a corresponding heterogeneous information network from the IMDb dataset. If users watched a certain movie and wrote a review about this movie, no matter whether they liked the movie or not, we say we observed the user-item interaction and we set this feedback as 1, otherwise, we set as 0. When building this dataset, we mapped two datasets using titles and release date of the movies, which could be erroneous on certain movies so the results we presented below are lower-bound of the actual performances.

Table 2: Meta-path examples (we set  $n = 1$  and 2. “biz” is short for “local business”)

Network	Meta-Path
IM100K	user – (movie – tag – movie) <sup>a</sup>
	user – (movie – director – movie) <sup>a</sup>
	user – (movie – genre – movie) <sup>a</sup>
	user – movie – plot – movie
Yelp	user – (biz – category – movie) <sup>a</sup>
	user – (biz – customer – biz) <sup>a</sup>
	user – biz – checkin – biz
	user – biz – location – biz

The second dataset is the Yelp challenge dataset<sup>a</sup>. This dataset contains both user reviews and local business information (an information network). When a user wrote a review for a restaurant, we set the feedback as 1, otherwise it would be set to 0. We summarize these two datasets in Figure 5(a) and the schema of which can be found in Figure 2. Notice that the Yelp dataset is much sparser than the IM100K dataset, so the performances of all methods decline accordingly. The distributions of the user feedback can be found in Figure 5.

Both datasets have timestamps with each user item interaction. We split the feedback matrix  $R$  of both datasets for each user into training and test based on timestamps, i.e., we use 80% of the “past” feedback to predict 20% of “future” feedback. In Yelp dataset, we have to filter out all the users who only have 1 review since we can not create training and test data for such users.

## 6.2 Competitors and Evaluation Metrics

We implement several widely deployed or the state-of-the-art recommendation approaches as comparison methods as follows:

- **Popularity:** Recommend the popular items to users.
- **Co-Click:** Estimate conditional probabilities between items and recommend items with an aggregated conditional probability calculated using the training data of the target user.
- **NMF:** Non-negative matrix factorization on  $R$ , details of which is discussed in Section 2.3
- **Hybrid-SVM:** Use SVM-based ranking function [13] to learn a global recommendation model with user implicit feedback and meta-paths based similarity measures [24].

We use **HeteRec-g** to denote the proposed global recommendation model and **HeteRec-p** to represent the personalized recommendation models derived from user subgroup recommendation models. We utilize 10 different meta-paths in each information network, including the most simple meta-path *user – item*, which means user preferences can only be propagated to items with observed positive feedback. We list some other meta-paths and / or attribute similarity measures in Table 6.1. For explicit feedback recommendation evaluation, measures like *root mean square error* (RMSE) are the standard evaluation metric. However, these metrics do not suit the definition of **implicit feedback problem**. In this study, we test all methods as ranking models and use the well-studied information retrieval metrics including precision-at-position and top-10 *mean reciprocal rank* (MRR, Equation (10)) to evaluate and compare the performance of these methods.

$$\text{MRR}_K = \frac{1}{m} \sum_{i=1}^m \left( \sum_{e \in \text{test}(u_i)} \frac{1}{\text{rank}(u_i, e)} \right) \quad (10)$$

<sup>a</sup>[http://www.yelp.com/dataset\\_challenge/](http://www.yelp.com/dataset_challenge/)

Table 3: Performance Comparison

Method	IM100K				Yelp			
	Prec1	Prec5	Prec10	MRR	Prec1	Prec5	Prec10	MRR
Popularity	0.0731	0.0513	0.0489	0.1923	0.00747	0.00825	0.00780	0.0228
Co-Click	0.0668	0.0558	0.0538	0.2041	0.0147	0.0126	0.01132	0.0371
NMF	0.2064	0.1661	0.1491	0.4938	0.0162	0.0131	0.0110	0.0382
Hybrid-SVM	0.2087	0.1441	0.1241	0.4493	0.0122	0.0121	0.0110	0.0337
HeteRec-g	0.2094	0.1791	0.1614	0.5249	0.0165	0.0144	0.0129	0.0422
HeteRec-p	<b>0.2121</b>	<b>0.1932</b>	<b>0.1681</b>	<b>0.5530</b>	<b>0.0213</b>	<b>0.0171</b>	<b>0.0150</b>	<b>0.0513</b>

### 6.3 Performance Comparison

The performance of all 6 methods in the two datasets can be found in Table 3.

Based on Figure 5, user feedback data follow power law distribution, *i.e.*, a very small number of items have interaction with a large number of users. Due to this property, recommending the popular items to users has a decent performance (MRR=0.1923 in IM100K). Co-click method, as one of the most widely deployed technique, achieves MRR = 0.2041 in IM100K and has a similar performance as the NMF method in Yelp (MRR=0.0371).

We implemented the NMF as the CF baseline (details of this method can be found in Section 2.3). We set the dimensionality of the low-rank representations  $d = 20$  in IM100K and  $d = 60$  in Yelp with cross validation in training dataset. We use the same method and settings in the diffusion-based latent feature generation method for the proposed approaches. With parameter tuning and additional information (*e.g.*, [9]), NMF may perform better than the results in Table 3. However, the same performance improvement can be achieved in our methods accordingly by replacing the NMF solver in Equation (3) with a more advanced technique. As presented in Table 3, NMF achieved MRR = 0.4938 and Prec1 = 0.2061 in IM100K dataset and MRR = 0.0382 and Prec1 = 0.0162 in Yelp dataset. This method outperforms other baselines methods in both datasets.

Hybrid-SVM method is a hybrid recommendation approach which uses the same amount of information as our proposed methods. This method combines both implicit feedback and heterogeneous relationship information following the intuitions of our study. However, it adopts an SVM based ranking framework [13] and uses PathSim [24] measures as features when defining the recommendation model. Without the proposed diffusion-based feature generation method, the learning algorithm and the recommendation model personalization, Hybrid-SVM can not fully take advantage of the feedback data and the heterogeneity of the information network. This method can only achieve MRR = 0.4493 in IM100K (compared to 0.4938 with NMF) and MRR = 0.0337 (compared to 0.0371 with Co-Click method). The low-performance of the Hybrid-SVM method proves the effectiveness of the proposed framework.

Our proposed global recommendation model (HeteRec-g), which takes advantage of both user feedback and the related information network, and users the same amount of information as Hybrid-SVM, beats all baseline methods in both datasets. It improves MRR by 6.1% compared to NMF in IM100K and 10.4% in Yelp dataset. This proves our assumption that adding information network as external knowledge with the proposed approach can alleviate the data sparsity issue and improve the recommendation quality. Moreover, HeteRec-g produces much more accurate recommendation results compared to Hybrid-SVM (in IM100K, MRR of HeteRec-g is 0.5249 while MRR of Hybrid-SVM is only 0.4493). Both methods utilize the same set of meta-paths, and use the same sample rate during training. Both methods define “global” recommendation models since they apply

the same model to all the users. The performance increase of HeteRec-g proves the effectiveness of our diffusion-based latent feature generation method. Another interesting observation is that the MRR gain of HeteRec-g compared with NMF in the relatively dense IM100K dataset is less than it is in the sparser Yelp dataset (6.1% v.s. 10.4%), which fits our intuition that when feedback dataset is sparser, the informative network-based recommendation approach can improve the performance even more. When training, we employ a uniform sample rate  $10^{-5}$  in SGD and we apply the same rate to all supervised approaches in this experiment. Parameter tuning of the sample rate is discussed later this section.

HeteRec-p as the personalized recommendation approach can further improve the performance in both datasets. HeteRec-p method clusters users based on their interests and utilizes personalized model parameters when recommending. This approach can distinguish user behaviors while HeteRec-g treats all users as the same. We use  $c = 10$  in IM100K dataset and  $c = 100$  in Yelp dataset. We discuss the strategy of choosing the correct granularity later this section. Compared with the global recommendation model (HeteRec-g), personalized models can provide higher quality recommendation results in both datasets. It improves Prec5 by 7.9% and MRR by 5.4% in IM100K, and improves Prec1 by 29% and MRR by 21.5% in Yelp. This verifies that different users indeed have different interests and behavior patterns. Applying global recommendation model to all users can not distinguish user differences properly. Recommendation models which are learned within user subgroups can represent user interests and behaviors at a finer granularity level, and thus HeteRec-p can generate better results. Compared with baseline methods which only utilize user feedback data, HeteRec-p surpasses NMF by 12.0% in terms of MRR in IM100K and beats the MRR score of Co-Click by 38.3% in Yelp dataset.

Overall, the proposed recommendation models outperform all comparison methods in both IM100K and Yelp datasets. The experiments verify that using heterogeneous information networks in recommender systems can improve recommendation quality. Personalized recommendation models can better distinguish user interests and behaviors, and thus can lead to personalized and more accurate recommendation results.

### 6.4 Performance Analysis

We analyze the performance of Co-Click, NMF, HeteRec-g and HeteRec-p in different recommendation scenarios. We ran the following analytical experiments on both datasets and observed similar performance change in both datasets. Due to the page limitation, we only present these findings with IM100K dataset.

We first study the correlation between the performance of different recommendation methods and the training feedback data size of each user. We split all users into 6 groups based on their individual training data size. Users in group 1 provided very limited number of feedback (average size is 13) while users in group 6 provided the most amount of



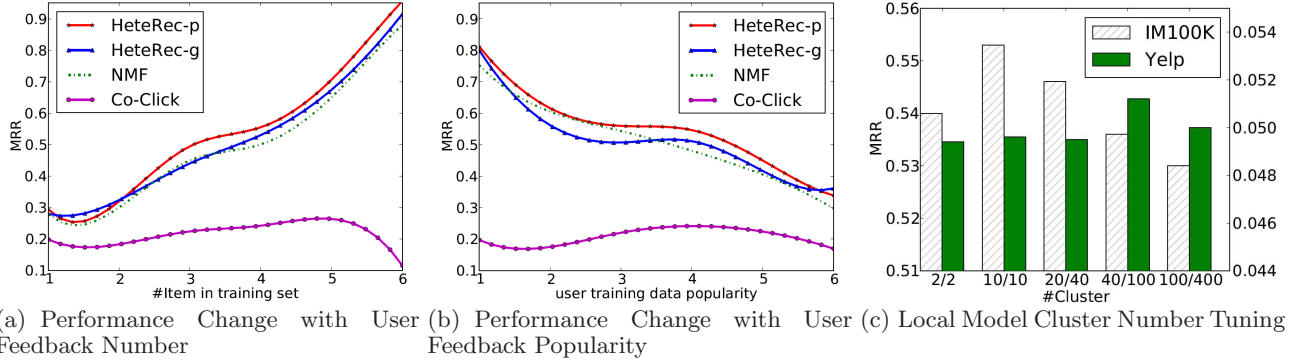


Figure 6: Performance Analysis and Parameter Tuning

feedback (average size is 224). We apply all 4 methods in each group. The results of this study can be found in Figure 6(a). One can notice that overall HeteRec-p outperforms all the other methods for users with different feedback sizes. When the user feedback size is small, *i.e.*, the data sparsity issue is severe, the performances of all methods are bad. When the feedback size of each user increases, the performances increase accordingly. The performance of Co-Click however does not change as much among different groups. This proves that all CF based methods could be affected when data are sparse. When feedback data are insufficient, it is critical to utilize information network as external knowledge to alleviate this issue.

We then study the correlation between performances of the 4 methods and the popularity of the items that users interacted in the feedback dataset. We split users into 6 groups based on the average popularity of the movies they interacted with in the training dataset. Users in group 1 prefer less popular movies (average popularity of items is 71) while users in group 6 prefer the most popular movies (average popularity is 281). The results of this study can be found in Figure 6(b). Similar to the previous study, overall HeteRec-p outperforms all the other methods for different user groups. Interestingly, all CF based methods perform better for users who prefer unpopular movies. This finding may be counter-intuitive. It would seem like popular movies are easier to handle since the related data are sufficient. However, users who prefer popular movies usually do not have specific interests (they watch anything popular without considering genres, stories or any other types of information). Recommending movies to such users is always challenging. One possible way of handling this problem is to identify such users and use popularity based methods for the recommendation.

## 6.5 Parameter Tuning

The proposed methods have several additional parameters compared with other methods.

In Equation (9),  $\lambda$  controls  $L_2$  regularization of the function. We cross-validated this parameter and set it to 0.1 when optimizing objective function. Another parameter is the sampling rate in SGD when estimating parameters. As mentioned in Section 5, with the proposed objective function, the scale of the training dataset is  $O(mn^2)$  which can be overwhelming in large datasets (this number in Yelp dataset is approximately  $10^{12}$ ). Instead of using the entire dataset, we only sample a subset during training. We study the relationship between the sample rate and the performance of HeteRec-g in IM100K (Figure 7).

Notice the x-axis in Figure 7 is at log scale. When sample rate is more than  $10^{-5}$ , the performance of the global

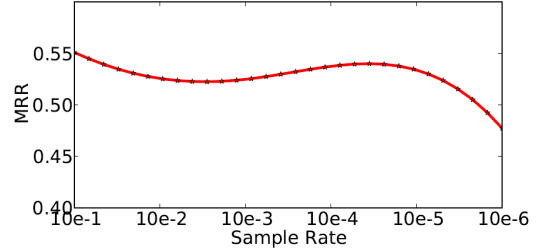


Figure 7: Sample Rate Tuning

model is relatively stable which means we do not need all the data to learn a high quality model with the proposed approach. However, inefficient training data size could harm the performance when we only supply  $10^{-6}$  or less training data.

Another parameter for HeteRec-p is the number of user clusters. As presented in Figure 6(c), although not very sensitive to this parameter, HeteRec-p does perform differently with different numbers of clusters. It peaks at  $c = 10$  in IM100K and  $c = 100$  in Yelp dataset compared with other parameters tested. When the number of clusters is small, HeteRec-p could not distinguish users behavior very well while a large cluster number could lead to training data deficiency for each subgroup model, and thus lead to performance decreasing.

## 7. RELATED WORK

### 7.1 CF Based Hybrid Recommender Systems

As one of the most popular recommendation approaches, collaborative filtering techniques have been extensively studied from different perspectives [23] [8]. Among different techniques, matrix factorization-based models [22, 14] are widely employed in many systems due to their good performances [15].

Recent studies on collaborative filtering turn to leveraging different types of external information so that data sparsity problem can be addressed and better recommending performance can be achieved. [20, 6, 1] incorporate item or user profiles (e.g., user demographic and item attributes) into collaborative filtering framework, leading to content-based collaborative filtering.

In contrast to content-based information, knowledge extracted from relational data are attracting increasing interest. In particular, several works leverage different social relationships in social network such as trust relationship [18,

10], friend relationship [19, 7], and user membership [31]. to boost collaborative filtering since it provides user's like-minded neighbors. Yu *et al.* [29] suggest the utilization of graph Laplacian regularization to exploit entity similarity measurements defined along different meta-paths.

To the best of our knowledge, existing relation-based collaborative filtering focuses on extracting knowledge from single or multiple homogeneous networks, i.e., networks with single type of nodes and links. Our work discusses leverage external knowledge extracted from heterogeneous information networks for collaborative filtering boosting at a personalized level.

On the other hand, regarding data characteristics, most existing methods handle user's explicit feedback like item ratings. Recently there has been increasing interest on usage of user implicit feedback to conduct collaborative filtering [9, 21], which are much easier to be collected.

## 7.2 Information Network Analysis

Heterogeneous information networks which contain multi-typed entities and links are the general data format of knowledge graphs. Information network analysis and mining have gained wide attention in both academia and industry [27]. Many researchers believe that the heterogeneity and rich-relation nature make information network a better data representation in many scenarios. A lot of information network mining and learning tasks have been done in the past couple of years, including clustering [25] [26], classification [12], and link prediction [28] [16], *etc.* Studies regarding entity similarity measurements, as a fundamental technique, have been actively engaged in many research works as well [4] [11] [24]. Researchers also discover that certain similarity measurements could be defined along paths in information networks, and such path compatible measurements could capture different similarity semantic meanings and can be used in different applications [17] [24]. These papers also motivated user guided or personalized data mining and analysis with information networks [25].

## 8. CONCLUSION AND FUTURE WORK

In this paper, we study recommendation in the scope of heterogeneous information networks. We propose a generic recommendation framework for implicit feedback dataset by taking advantage of different types of entity relationships in heterogeneous information networks. We define recommendation models at both global and personalized levels. Personalized recommendation models can be efficiently generated on the fly, and this approach can provide high quality personalized recommendation results compared to other recommendation methods. A Bayesian ranking process is utilized to estimate the weights of the recommendation models. We compared the proposed approaches with several widely employed or state-of-the-art implicit feedback recommendation techniques, and empirical study demonstrates the effectiveness of our methods. We also analyzed the performance of these methods under different scenarios and explained the reasons of the performance drift. Interesting future work includes on-line recommendation model update with users providing model feedback, large scale recommendation model in information networks as well as approximate learning process with low time complexity.

## Acknowledgments

The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and W911NF-11-2-0086 (Cyber-Security), the U.S. Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, DTRA, DHS, and U.S. Na-

tional Science Foundation grants CNS-0931975, IIS-1017362, IIS-1320617, IIS-1354329.

## 9. REFERENCES

- [1] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *ICML*, 2004.
- [2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 2010.
- [3] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [4] S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *WWW*, 2007.
- [5] C. H. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *PAMI*, 2010.
- [6] Q. Gu, J. Zhou, and C. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, 2010.
- [7] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *RecSys*, 2009.
- [8] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR*, 2003.
- [9] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.
- [10] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *SIGKDD*, 2010.
- [11] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *SIGKDD*, 2002.
- [12] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *SIGKDD*, 2011.
- [13] T. Joachims. Training linear svms in linear time. In *SIGKDD*, 2006.
- [14] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 2008.
- [15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [16] T.-T. Kuo, R. Yan, Y.-Y. Huang, P.-H. Kung, and S.-D. Lin. Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In *SIGKDD*. ACM, 2013.
- [17] N. Lao and W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 2010.
- [18] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, 2009.
- [19] H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.
- [20] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI*, 2002.
- [21] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UIAI*, 2009.
- [22] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.
- [23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.
- [24] Y. Sun, J. Han, X. Yan, S. P. Yu, and T. Wu. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. In *VLDB*, 2011.
- [25] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu. Integrating meta-path selection with user guided object clustering in heterogeneous information networks. In *KDD*, 2012.
- [26] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, 2009.
- [27] M. Szell, R. Lambiotte, and S. Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636–13641, 2010.
- [28] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *SDM*, 2012.
- [29] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. In *IJCAI HINA*, 2013.
- [30] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han. Entity recommendation in heterogeneous information networks with implicit user feedback. In *RecSys*, 2013.
- [31] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys*, 2011.