LEVERAGING HETEROGENEOUS INFORMATION NETWORKS FOR PERSONALIZED ENTITY RECOMMENDATION

BY

BRANDON NORICK

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Jiawei Han

Abstract

Recommendation is a challenging but important task which has applications in nearly every sector of industry as well as in academia. There are a wide variety of approaches to the recommendation problem, with network-based techniques garnering increasing interest and study in recent years. However, most of these studies only explore the problem in the context of a single relationship between entities, such as a following relationship in a social network like Twitter. Such approaches ignore the complex environment in which most recommendation tasks exist in favor of simplifying the problem. The complexity of human decision making necessitates approaches which can utilize the heterogeneous environments in which the recommendation task is set rather than reducing them to single relationship.

In this work, we explore the problem of entity recommendation without such a simplification, instead we utilize *heterogeneous information networks* to capture the complexity of the behaviors for which we are seeking to make recommendations. Our proposed approach captures the different behaviors of individuals by examining their heterogeneous relationships in the network and as a result can provide high-quality personalized recommendations from implicit feedback represented in heterogeneous information networks.

We begin by introducing meta-path-based latent features, which capture the connectivity of entities in the network along different paths, giving us a foundation which explicitly accounts for the heterogeneous nature of the task. Upon this foundation we build a global recommendation model using a ranking optimization technique known as Bayesian Personalized Ranking. We extend this global model into a personalized model, building a model which can capture the differences present in the network that describe the preferences of different users. Finally, empirical studies show that our techniques are more effective than several popular and state-of-the-art entity recommendations techniques.

To my parents, for their love and support, and my wife for her patience.

Acknowledgments

Many thanks to my adviser, Professor Jiawei Han, who has been ever supportive of me throughout my research career. From my first visit to Champaign-Urbana, I have felt his kindness and encouragement, as well as learned from his academic guidance.

I would also like to thank the staff of the Computer Science Department at UIUC. You were always there for me throughout the years and I will forever be grateful. In alphabetical order by last name, thank you to: Holly Bagwell, Steve Herzog, Mary Beth Kelley, Candy Foster, Kara MacGregor, and Kathy Runck.

I would also like to acknowledge my many collaborators, especially Huan Gui, Fangbo Tao, and Jialu Liu. I sincerely appreciate all of the help you have given me over the years and enjoyed all of the stimulating conversations we had surrounding our research.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

G, G_S	Heterogeneous information network (HIN) and network schema.
$\mathcal{A}, \mathcal{R}$	Set of entity types and set of relation types for a HIN.
R	User implicit feedback matrix.
m, n	Number of users and number of items.
U, V	Low-rank representations of users and items.
d	Dimensionality of the latent feature space.
u, e	A user and item (i.e., entity).
$\mathcal{P}$	A meta-path.
p	A specific path instance following some meta-path.
$\tilde{R}^{(q)}$	Preference diffused matrix along the qth meta-path.
L	Number of meta-paths.
$\hat{U}^{(q)}, \hat{V}^{(q)}$	Low rank representations of users and items when user preference is diffused along the qth meta-path.
$\theta, \theta^{\{\cdot\}}$	Global and personalized model parameters.
C	User clusters.

Chapter 1

Introduction

As the world becomes more connected and services we increasingly move online, users are faced with a problem: there are too many choices available in nearly every facet of their lives. Many methods have been proposed for enabling users to filter and discover information that is both interesting and useful for their particular needs. Perhaps the most important of these filtering and discovery methods is entity recommendation, due to its effectiveness and widespread adoption. The effectiveness of this group of techniques is largely thanks to the abundance of user feedback data, both implicit interaction data and explicit rating data. In contrast to early recommendation techniques which only considered user feedback, more recent hybrid recommender systems combine both user feedback and additional information about users or items to achieve better recommendation results in certain scenarios [6, 21].

Given the data-rich space in which most modern applications for recommendation exist, additional contextual information about the users and entities being recommended can be exploited to improve recommendation performance. For example, it may be desirable to use a recommendation technique which can leverage user demographic information, product details, or location data about online content such as blog posts. The entity recommendation problem tends to exist in an environment that can be expressed as a heterogeneous information network (HIN) containing different types of entities and a variety of relationships between them. A concrete example for the movie recommendation problem is given in Figure 1.1. For a classical recommendation technique, only user and movie entities would be considered, and the only relationship used would be the user-movie relationship. However, even in this simple example we can see that there is an abundance of other data in the form of other entity types and other relationships. Entities such as directors, actors, and genre can be linked to movies with relationships which express their role in the movie, e.g., directed or acted-in. This additional information can potentially be leveraged by a hybrid recommender system to provide better recommendations.

Previous studies have found that utilizing additional user or item relationship information can lead to higher quality recommender systems. Our technique follows the hybrid recommender system model with key differences. While most previous link-based hybrid approaches have only leveraged a single type of relationship, e.g., friend relationship [20], trust relationship [11], or user membership [32], we propose to

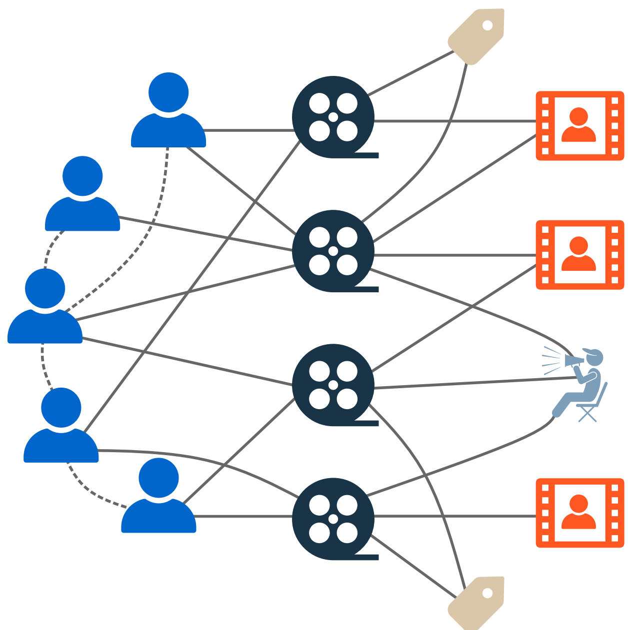


Figure 1.1: An example of a heterogeneous information network which may be used for movie recommendation. Users are linked by a social network (dashed lines), their implicit feedback over movies is captured in the relationship depicted between the users and movies, and different entities related to each movie are depicted on the right. The other entities in this network include actors, directors, and tags.

study the problem in the context of heterogeneous information networks and develop a technique which can use the diverse data available to achieve superior recommendation performance. We also improve upon the personalization approach taken in previous studies [30, 31], which apply the same model to all users when recommending items and rely on the user feedback data to achieve personalization. While such an approach may work in simple cases, it is not powerful enough to fully capture user interests and preferences and may lead to undesirable recommendations. For example, two users who watched the same movie may have done so for entirely different reasons. One may have watched it because they are a fan of the genre and were interested in the story, whereas the other may have watched it due to recommendations from their friends. If we simply apply the same model to both users without accounting for this difference, the recommendation results may only satisfy the information discovery needs of users which fit into the model we have chosen. We propose other personalization approaches which more closely model user preference and can capture different interests and preferences across different users.

In this work, we propose a novel entity recommendation technique which exploits the rich data available in heterogeneous information networks. Our technique relies on the combination of implicit user feedback and a variety of other entities and relationships between them to produce entity recommendations. Personalization is achieved through both the use of user implicit feedback and building personalized models for different users in order to capture their diverse interests and preferences.

At a high level, we take advantage of the heterogeneous information network and the diverse data therein by diffusing the observed user feedback along different meta-paths representing different preferences. This results in a set of possible recommendation candidates which are related to the preference represented to the meta-path used to generate them. We apply matrix factorization techniques to these diffused versions of the user feedback, resulting in user and entity latent representations which we can use as the foundation of the recommender system. We first combine these latent representations to learn a global model. To further personalize the model, we use the latent features to build different models for different users, thereby capturing the diverse preferences that users express. To estimate each model we utilize a Bayesian ranking optimization technique [22]. We perform empirical studies in two real-world datasets, IMDb-MovieLens-100k and Yelp, which show that the proposed approaches outperform several popular and state-of-the-art implicit feedback recommender systems.

The remainder of this thesis is organized as follows. The related work is discussed in more detail in Chapter 2. Requisite background and preliminaries are introduced in Chapter 3. Our methodology, including both the global and personalized techniques, is discussed in Chapter 4. Our experiments and analysis are presented in Chapter 5, and finally conclude in Chapter 6.

Chapter 2

Related Work

2.1 Collaborative Filtering Based Hybrid Recommender Systems

Collaborative filtering is one of the most widely used techniques for recommendation and has been studied at length from many angles [24, 9]. Matrix factorization based approaches [23, 15] tend to be favored due to their strong performance [16].

Recently, research has focused on the use of extra information in addition to the explicit or implicit user feedback as a way to combat data sparsity and improve performance. Several works involve either entity or user information into their frameworks which is sometimes referred to as content-based collaborative filtering [21, 6, 1].

Other techniques aim to solve the problem using a link-based approach. As this approach has gained in popularity, there have been works leveraging different social relationships in social networks such as friendship [20, 8], trust [19, 11], and group membership [32]. The end goal is to improve performance by exploring a user's similar neighbors from these aspects. One study [20] suggests the use of graph Laplacian regularization to leverage entity similarity along defined meta-paths which can capture different semantic relationships. While these approaches focus on learning from one or more homogeneous networks, our work explores the use of heterogeneous information networks which are richer and contain many more semantically meaningful relationships. Through their use we introduce a framework for collaborative filtering which not only has the ability to exploit an immense amount of context but also allows for further personalization.

With regard to the user feedback data being consumed, most previous work focused on explicit user feedback such as item ratings on a predefined rating scale. However, this data is difficult to collect, so implicit feedback approaches for which the data is much easier to collect have been receiving more attention recently [10, 22].

2.2 Information Network Analysis

Heterogeneous information networks, those which are made up of multiple types of entities and relationships, are flexible enough to describe the complex world in which we live, similar to knowledge graphs. In both industry and academia, the mining and analysis of information networks has gained widespread attention [28]. Their heterogeneity and ability to represent complex relationships makes them a perfect candidate for modeling the real world. Studies have focused on many classical data mining techniques for heterogeneous information networks, such as clustering [26, 27], classification [13], and link prediction [17, 29]. The exploration of entity similarity measures in these networks has also been studied [4, 12, 25]. The use of path-based similarity measures, which are flexible enough to meet the demands of any application and can capture complex and semantically meaningful relationships has been shown to be effective [18, 25]. These works also led to user-guided approaches to common problems like clustering [26].

Chapter 3

Background and Preliminaries

This chapter will cover the requisite background knowledge needed to understand our technique, as well as introducing the problem definition and other preliminaries.

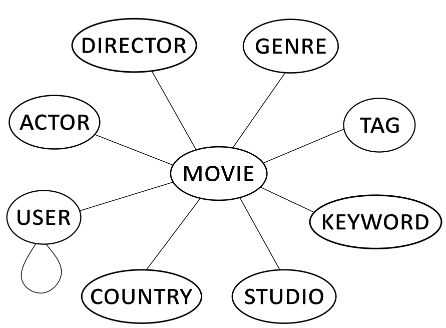
3.1 Heterogeneous Information Networks

We define information networks in the following way, which follows [25],

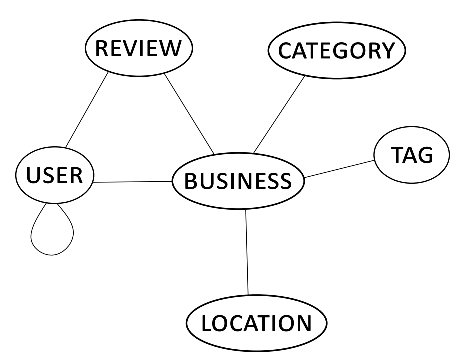
Definition 3.1 (Information Network)**.** An information network is defined as a directed graph $G = (V, E)$ with an entity type mapping function $\phi : V \to \mathcal{A}$ and a link type mapping function $\psi : E \to \mathcal{R}$. Each entity $v \in V$ belongs to an entity type $\phi(v) \in \mathcal{A}$, and each link $l \in E$ belongs to a relation type $\psi(l) \in \mathcal{R}$.

For an information network $\mathcal{I}$, if $|\mathcal{A}| > 1$ or $|\mathcal{R}| > 1$ then we refer to the network as a *heterogeneous information network (HIN)*. In order to remain consistent with previous works on recommendation, we will refer to the entities being recommended as *items*.

As a way to abstract HINs, we represent them in an abstract graph where the entity types are nodes and they are connected by the relations present in the network. This abstract graph, also known as a *network schema*, is similar to entity-relation diagrams used to represent relational databases. We will denote such network schemas as $G_S = (\mathcal{A}, \mathcal{R})$. Examples of simple network schemas can be found in Figure 5.3.



(a) IMDb Network



(b) Yelp Network

Figure 3.1: Example network schemas for two possible heterogeneous information networks.

3.2 Implicit User Feedback

Implicit feedback from users is represented in a binary matrix $R \in \mathbb{R}^{m \times n}$, where the m users are the rows and the n columns are the items over which the implicit feedback is being collected. The values in the matrix R can be described as follows:

$$R_{ij} = \begin{cases} 1, & \text{if } u_i \text{ interacted with } e_j \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Importantly, the value 1 in R represents an interaction between a user and an entity, rather than an explicit piece of feedback from the user. For example, this could mean a user clicked on a link in a search engine, watched a movie on an online service such as Netflix, or browsed a page for a particular restaurant on Yelp. A value of 1 in R does not necessarily mean that a user liked the entity with which they interacted. Indeed, it is not uncommon to watch a movie which you find interesting, only to discover that you disliked it. Along the same lines, a value of 0 in R does not mean that a user dislikes a particular entity. Rather, the 0 values in R are a mixture of three types of relationships: entities which the user certainly dislikes and thus has chosen not to interact with, entities which the user is uninterested in, and entities which the user has not yet discovered. It is worth noting that several previous studies have included additional assumptions about the implicit feedback, such as interaction frequency assumptions. In order to keep this study focused, we do not explore these avenues, but methods along these lines may be added to the proposed models in a fashion similar to previous works if desired.

3.3 Implicit Feedback Matrix Factorization

Factorization of the implicit feedback matrix has been studied in prior works [5, 10], where low-rank matrices are learned to represent users and items in order to approximate the feedback matrix through their combination. More specifically, factorization techniques aim to approximate the implicit feedback matrix R as follows:

$$R \approx UV^T \tag{3.2}$$

where $U \in \mathbb{R}^{m \times d}$ are the feature representations of users in some latent space, and $V \in \mathbb{R}^{n \times d}$ are the feature representations of items in some latent space. To satisfy the low-rank constraint such techniques require that $d < \min(n, m)$.

Given low-rank matrices U and V, a score between u_i and e_j can be computed according to $r(u_i, e_j) = U_i V_j^T$, where U_i is the ith row of the matrix U and V_j is the jth row of the matrix V. For each user u_i, items can be sorted according to these scores and we can recommend the top-k items which u_i has no implicit feedback for (i.e., items which u_i has not yet interacted with).

In order to find matrices U and V which best solve Equation 3.2, non-negative matrix factorization (NMF) techniques like that discussed in [5] can be applied to the implicit feedback matrix R. Other approaches have also been studied [6, 10, 20], which improve performance by incorporating extra information.

In this work we propose models which employ matrix factorization to learn a set of user and item features which capture different semantic relationships. Importantly, the proposed models are orthogonal to the choice of matrix factorization technique and one could employ our models along with more advanced factorization approaches. To remain focused, in this study we use the NMF method in [5] to learn the features which form the basis for our models. However, due to the orthogonality between the proposed models and the factorization approach, our methods can be improved further through the use of more advanced factorization techniques.

3.4 Problem Definition

The implicit-feedback based recommendation problem which we study in this work is defined as follows:

Definition 3.2 (Problem Definition)**.** Given a heterogeneous information network G with user implicit feedback matrix R, or a user u_i we aim to build a personalized recommendation model which can recommend a ranked list of items that are of potential interest to u_i.

The symbols used in this chapter and the remainder of this thesis can be found in the List of Symbols preceding Chapter 1.

Chapter 4

Methodology

In this chapter we present our proposed recommendation models. Through the incorporation of heterogeneous information networks into the user-item latent feature learning we capture both the implicit feedback along with user preference over other complex relationships present in the information networks. This is accomplished by diffusing user preferences along different semantic relationships in the network, combining the implicit user feedback along with other, more complex, relationships which can be described by heterogeneous information networks. Using these enriched latent features, we propose a scoring function for recommendation which can be applied to all users. We refer to this as a *global* model, the details of which are covered in Section 4.2. In the context of this model, *global* means that it is applied in the same way to all users despite their potentially different preferences. However, this model is still based on user implicit feedback, so the recommendations for different users will not be the same. Extending this approach, we present our personalized recommendation model in Section 4.3 which allows the model to learn different users' preferences. Finally, the learning algorithms for the proposed techniques are discussed in Section 4.4.

4.1 Motivation

4.1.1 Meta-Paths

In an information network, be it homogeneous or heterogeneous, the entity recommendation task can be solved by finding highly connected entities for a given user. In heterogeneous information networks, two nodes can be connected by different paths which may be wildly different. Two paths connecting the same nodes may be composed of extremely different entity types, relation types, and may be of completely different lengths. Consider, for example, users u_1 and u_2 in Figure 4.1, and the paths $u_1 \to u_3 \to m_3 \to d_1 \to m_2 \to a_3 \to m_1 \to u_2$ and $u_1 \to m_2 \to a_1 \to m_3 \to u_2$. As proposed in [25], we introduce the concept of a *meta-path* to describe the widely varying paths in heterogeneous information networks. A meta-path is defined using the network schema of a HIN and describes a particular way in which two entity types could be connected in the network.

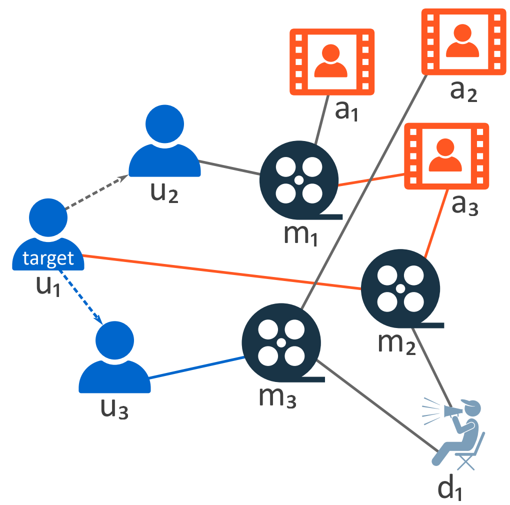


Figure 4.1: A simple heterogeneous information network highlighting two meta-paths.

Definition 4.1 (Meta-Path)**.** A meta-path $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \ldots \xrightarrow{R_k} A_k$ is a path in a network schema $G_S = (\mathcal{A}, \mathcal{R})$, which defines a new composite relationship $R_1 R_2 \ldots R_k$ between entity types A_0 and A_k, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$ for $i = 0, \ldots, k$, $A_0 = dom(R_1) = dom(\mathcal{P}), A_k = range(R_k) = range(\mathcal{P})$, and $A_i = range(R_i) = dom(R_{i+1})$ for $i = 1, \ldots, k-1$.

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

We differentiate between explicit path instances and meta-paths by using the notation p and $\mathcal{P}$ respectively. Given the above definition, one can see that any path p in a heterogeneous information network will follow an associated meta-path $\mathcal{P}$. Prior work shows that meta-paths can be used for a variety of semantically meaningful purposes, such as entity similarity semantic disambiguation [24, 28]. Consider the following example as motivation for the use of meta-paths to help solve the entity recommendation problem.

Example 4.1 (Meta-paths in IMDb)**.** The graph schema of IMDb defined in Figure 3.1a allows us to define many different meta-paths between users and movies, each of which represents a more complex semantic relationship. Two possible meta-paths and their associated semantically interesting meaning are as follows:

$$\mathcal{P}_1 = user \xrightarrow{follows} user \xrightarrow{watched} movie$$

(movies watched by users which are followed by a user)

$$\mathcal{P}_2 = user \xrightarrow{watched} movie \xrightarrow{performed-in^{-1}} actor \xrightarrow{performed-in} movie$$

(movies containing an actor which the user has seen in another movie)

Example path-instances following these meta-paths can also be found in Figure 4.1, where we used a blue lines to represent $\mathcal{P}_1$ and orange lines to represent $\mathcal{P}_2$. The two semantic relationships are very different:

movies highly connected to a target user by $\mathcal{P}_1$ will be those which have been watched my many users which are in the target user's social network, whereas movies which are highly connected to the target user along $\mathcal{P}_2$ will be ones which contain many actors which the user has seen in other movies. By measuring user-movie proximity along these and other meta-paths, we can capture the inherent variety in user preference and make recommendations which make are more effective.

When the relationship type between two entity types is not ambiguous (i.e., there is just one relationship type connecting the two) then it may be omitted for a simpler notation. Additionally, repeated parts of a meta-path may be compressed using an exponent. For example, the meta-path

$$user \xrightarrow{watched} movie \xrightarrow{in} genre \xrightarrow{in^{-1}} movie \xrightarrow{in} genre \xrightarrow{in^{-1}} movie$$

can be simplified to *user*—(*movie*—*genre*—*movie*)^2.

4.1.2 Preference Diffusion

Given the building blocks of implicit user feedback data as described in Section 3.2 and meta-paths as defined above, we can now introduce the user preference diffusion approach. In this context, the term *user preference* is used to mean the user interests which motivate the implicit feedback data. Remember that in implicit feedback data, a value of 1 means that a user is more interested in that item than other items with a value of 0. As a result, if we can view the implicit feedback from a variety of semantic perspectives and find similar items to ones in which the user is interested under each of these different perspectives, then we can make entity recommendations using these semantically meaningful views of the implicit feedback.

Consequently, we focus on meta-paths of the form *user*—*item*—$(\cdots)$—*item* in our recommendation models. Such meta-paths will diffuse the user implicit feedback data to items which the user may not be directly connected. As a result we can measure the relatedness between a user and all possible items along each semantically meaningful meta-path, allowing us to build a model with proposes unobserved user-item interactions which appear to be the most meaningful from a variety of different perspectives of the implicit feedback. This meta-path based diffusion approach captures user preference which are hidden in the implicit feedback data, and is therefore referred to as preference diffusion.

In order to actually measure the relatedness between a user and any item when the feedback is diffused along some meta-path $\mathcal{P} = R_1 R_2 \ldots R_k$. We extend PathSim, as proposed in [25], to measure the preference of user i for item j diffused along $\mathcal{P}$ as follows:

$$s(u_i, e_j|\mathcal{P}) = \sum_{e \in \mathcal{I}} \frac{2 \times R_{u_i,e} \times |\{p_{e \rightsquigarrow e_j} : p_{e \rightsquigarrow e_j} \in \mathcal{P}'\}|}{|\{p_{e \rightsquigarrow e} : p_{e \rightsquigarrow e} \in \mathcal{P}'\}| + |\{p_{e_j \rightsquigarrow e_j} : p_{e_j \rightsquigarrow e_j} \in \mathcal{P}'\}|} \tag{4.1}$$

where $\mathcal{P}' = R_2 \dots R_k$ and $p_{x \rightsquigarrow y}$ is a path instance between nodes x and y. Additionally, the notation $|\{p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in \mathcal{P}'\}|$ represents the number of path instances from x to y in the network which follow the meta-path $\mathcal{P}'$. Remembering the types of paths which we have chosen to consider, $dom(\mathcal{P}') = item$ and $range(\mathcal{P}') = range(\mathcal{P}) = item$.

Equation 4.1 models the user preference diffusion with two components: (1) the observed user implicit feedback of user u_i on each item e, $R_{u_i,e}$, and (2) the meta-path based proximity of the items in which the user has shown interest and all possible items e_j, captured by the paths $p_{e \rightsquigarrow e_j}$. Similar to PathSim, the number of paths between e and e_j is normalized so as not to favor very highly connected (i.e., popular) entities unfairly. A toy example of the preference diffusion process is shown in Figure 4.2 and described below.

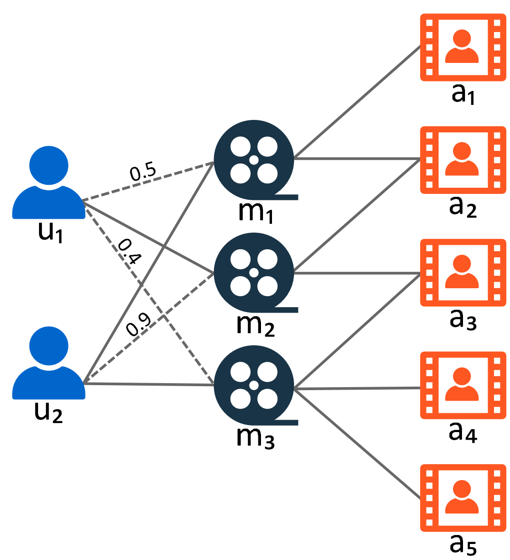


Figure 4.2: A toy example including preference diffusion scores calculated along the meta-path $\mathcal{P} = user—movie—actor—movie$. Solid lines are observed user implicit feedback, while dashed lines are the diffused user preferences.

Example 4.2 (Preference Diffusion)**.** For this toy example we use a small HIN containing two users, three movies, and five actors with the entities connected according to Figure 4.2. Of the six possible user-movie interactions, we can see that only three have been observed as implicit user feedback. Using the meta-path $\mathcal{P} = user—movie—actor—movie$, we calculate the preference diffusion of both users for the unobserved user-movie relations.

In this example, $\mathcal{P}' = \textit{movie—actor—movie}$, accordingly:

$$|\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}| = 2$$
$$|\{p_{m_2 \rightsquigarrow m_1} : p_{m_2 \rightsquigarrow m_1} \in \mathcal{P}'\}| = 1$$
$$|\{p_{m_2 \rightsquigarrow m_2} : p_{m_2 \rightsquigarrow m_2} \in \mathcal{P}'\}| = 2$$
$$|\{p_{m_3 \rightsquigarrow m_1} : p_{m_3 \rightsquigarrow m_1} \in \mathcal{P}'\}| = 0$$
$$|\{p_{m_3 \rightsquigarrow m_3} : p_{m_3 \rightsquigarrow m_3} \in \mathcal{P}'\}| = 3$$

We know from the user implicit feedback represented in the graph that u_1 did not watch m_1, so we can calculate the preference diffusion for (u_1, m_1) along $\mathcal{P}$ using Equation 4.1 as follows:

$$\begin{aligned}
s(u_1, m_1) &= \tfrac{2 \times R_{u_1,m_1} \times |\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}|}{|\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}| + |\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}|} + \\
&\quad \tfrac{2 \times R_{u_1,m_2} \times |\{p_{m_2 \rightsquigarrow m_1} : p_{m_2 \rightsquigarrow m_1} \in \mathcal{P}'\}|}{|\{p_{m_2 \rightsquigarrow m_2} : p_{m_2 \rightsquigarrow m_2} \in \mathcal{P}'\}| + |\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}|} + \\
&\quad \tfrac{2 \times R_{u_1,m_3} \times |\{p_{m_3 \rightsquigarrow m_1} : p_{m_3 \rightsquigarrow m_1} \in \mathcal{P}'\}|}{|\{p_{m_3 \rightsquigarrow m_3} : p_{m_3 \rightsquigarrow m_3} \in \mathcal{P}'\}| + |\{p_{m_1 \rightsquigarrow m_1} : p_{m_1 \rightsquigarrow m_1} \in \mathcal{P}'\}|} \\
&= \tfrac{2 \times 0 \times 2}{2+2} + \tfrac{2 \times 1 \times 1}{2+2} + \tfrac{2 \times 0 \times 0}{3+2} \\
&= 0.5
\end{aligned}$$

The other diffusion scores can be calculated similarly.

Calculating the preference diffusion score along a meta-path $\mathcal{P}$ for every user-item pair (u_i, e_j) will result in a user preference matrix $\tilde{R} \in \mathbb{R}^{m \times n}$ which describes the preference of each user viewed through the lens of the semantic relationship represented by $\mathcal{P}$. For example, if the $\mathcal{P} = \textit{user—movie—actor—movie}$ then $\tilde{R}_i$ (i.e., the ith row of $\tilde{R}$), represents the predict amount to which of user u_i would enjoy each movie if they prefer movies with actors that they commonly watch.

If we define L different meta-paths and calculate L different diffused user preference matrices (denoted $\tilde{R}^{(1)}, \tilde{R}^{(2)}, \ldots, \tilde{R}^{(L)}$) as above, then we can capture user preference over the items by combining the scores from these diffused preference matrices in a way which best describes the observed user feedback. Such a process is analogous to how users actually discover information and make decisions, weighing a variety of different preferences in their final decision to interact with an item or not.

4.2 Global Recommendation Model

Using the proposed preference diffused matrices $\tilde{R}^{(q)}$ for $q = 1, \ldots, L$ we can derive L low-rank user and item matrices using NMF, as discussed in Section 3.3. These low-rank matrices represent latent features for user and items under the L different semantic relationships described by the meta-paths corresponding to each diffused preference matrix. More specifically, using NMF we factorize the diffused preference matrix $\tilde{R}^{(q)}$ as follows:

$$\begin{aligned}(\hat{U}^{(q)}, \hat{V}^{(q)}) = \mathrm{argmin}_{U,V} ||\tilde{R}^{(q)} - UV^T||_F^2 \\ \text{s.t. } U \geq 0, V \geq 0\end{aligned} \tag{4.2}$$

where $\hat{U}^{(q)} \in \mathbb{R}^{m \times d}$ are the user latent features and $\hat{V}^{(q)} \in \mathbb{R}^{n \times d}$ are the item latent features under the qth meta-path. As with other low-rank techniques, $d < \min(m, n)$. As mentioned in Section 3.3, we apply the simplest NMF technique to solve Equation 4.2 rather than add complexity with a more involved technique, leaving that as an orthogonal direction in which performance may be improved if necessitated by the application.

By repeating this factorization for all L preference diffused matrices we obtain L pairs of latent features for users and items, $(\hat{U}^{(1)}, \hat{V}^{(1)}), \ldots, (\hat{U}^{(L)}, \hat{V}^{(L)})$. Each pair represents the latent features of users and items under the particular semantic relationship expressed by the corresponding meta-path as a result of the preference diffusion process. Intuitively, different relationships may have different levels of importance, meaning that a recommendation model should weight different feature pairs differently to best capture user preference. For example, knowing which actors perform in a movie is likely to have a stronger influence on a user's preference for a movie than knowing the studio which produced the movie. Accordingly, as in [31], we define the global recommendation model as follows:

$$r(u_i, e_j) = \sum_{q=1}^{L} \theta_q \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \tag{4.3}$$

where θ_q is a learnable weight for the predicted user and item interaction derived from qth preference diffused matrix, i.e., under the semantics of the relationship described by the qth meta-path. Due to the non-negativity of the latent features, we also enforce $\theta_q \geq 0$ as an optimization constraint.

Using the global recommendation model defined by Equation 4.3, we can generate scored for all user-item pairs and then, for each user, return the items with the top-k highest scores. We discuss model learning in Section 4.4.

4.3 Personalized Recommendation Model

Having proposed a global model which combines user implicit feedback with heterogeneous information networks through the use of user preference diffused latent features derived from different meta-paths, in this section we introduce personalization strategies which eliminate the built-in assumption that all users should be treated identically. When recommending items to users, the global model is applied in the same way to all users. Despite using each user's implicit feedback, the model itself does not distinguish different user interests and recommend items to users in different ways based on the interests they have exhibited in the implicit feedback data. For example, the global model may learn that, in general, users prefer to watch movies directed by famous directors. However, this rule certainly does not hold for all users, and it was only learned because overall it provided the best one-size-fits-all performance.

The global model serves as a good starting point for the personalized models, especially in the use of meta-path based user preference diffused matrices. We propose two extensions to the global model, making it more granular and capable of modeling different users' interests. At a high level, we do this by learning different recommendation models for different users enabling us to better capture user preferences by not treating all users identically.

Perhaps the most straightforward and expressive way to achieve the personalization goal would be to learn one model for each user using Equation 4.3, based only on their own implicit feedback. In this manner, each user's preference over the different semantic relationships present in the hetereogeneous information network would be captured. Unfortunately, the user feedback data follows a power law distribution as shown in Section 5.1, meaning that most users lack sufficient data on which to learn a personalized model.

One way to solve this data-sparsity problem is by grouping similar users into several groups, then learning a model for each group. The intuition underlying this approach is that, while different users may have different preferences from one another, there should be subgroups of users which share similar preferences. For example, there may be a group of older users who love traditional western movies, while there is another group of younger users who watch every movie with Brad Pitt. For the first group, a model will capture their collective preference for movies of a particular genre, whereas for the second group a model will capture their preference for movies with particular actors.

Following this logic, we propose to cluster users based on their interests and then learn a recommendation model for each cluster. A side effect of this approach is that if a user can be in multiple clusters, their personal interests can be modeled as a mixture of the interests of the clusters to which they belong, bringing us back to our original goal of a personalized model for each user. The personalized recommendation function for a user u_i is defined as follows:

$$r^*(u_i, e_j) = \sum_{k=1}^{c} sim(C_k, u_i) \sum_{q=1}^{L} \theta_q^{\{k\}} \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \tag{4.4}$$

where C_k represents the kth user cluster and $sim(C_k, u_i)$ is a measure of the degree to with user u_i is a member of cluster C_k. The learnable weights $\theta_q^{\{k\}}$ are analogous to those of the global model, but are learned across all of the c clusters so as to capture the different preferences of different users. The personalized model parameters are therefore $\theta^{\{\cdot\}} = \{\theta^{\{1\}}, \theta^{\{2\}}, \ldots, \theta^{\{c\}}\}$ of which there are $c \times L$, in contrast to the L parameters in the global model. This larger number of parameters enables us to better capture preferences at the user level, rather than the aggregate global level. In the following sub-sections we propose two different approaches which use this personalization strategy. Model learning for the personalized models is discussed in Section 4.4.

After estimating all the parameters needed to describe the model, $\theta^{\{\cdot\}}$, we can make user specific recommendations. For user u_i, we first determine how well their interests are described by each cluster using $sim(C_k, u_i)$, then we combine the recommendation models of each cluster according to the degree to which each cluster represents the user's preferences using Equation 4.4. As before, we return as recommendations the top-k items with the highest scores.

Choosing the number of clusters, c, can have a large impact on performance. Too few clusters may inhibit the personalized model from being able to distinguish user preferences well, similar to the global model. Too many clusters and the data sparsity issue returns due to the small number of user per cluster. One effective approach for determining a value for c is to use cross-validation in the training data to select a value. The impact of the number of clusters on the overall performance of the personalize models is discussed in Section 5.5.

4.3.1 Implicit Feedback Based Personalization

For the implicit feedback based personalization technique, we start from the user implicit feedback matrix R. Intuitively, this matrix captures the different preferences of different users as expressed by their interaction behavior. If we could cluster users according to their implicit feedback, each cluster could capture different user preferences as demonstrated by their behavior. However, due to the sparsity of the matrix R, clustering users directly is ineffectual. First, we learn low-rank but dense representations for users by applying NMF to R, giving us representations for each user. We then apply the well-studied and straightforward *k-means* algorithm to these user representations to cluster the users into c clusters using cosine distance as the distance metric between users. For consistency, $sim(C_k, u_i)$ is defined to be the cosine similarity between the low-rank representation of u_i and the cluster centroid for cluster C_k.

4.3.2 Heterogeneous Information Network Based Personalization

While the personalization strategy proposed in Section 4.3.1 is straightforward, it relies on matrix factorization of the implicit feedback matrix to capture complex user preferences and enable user clustering. This approach neglects all of the heterogeneous context data available in the HIN. A more principled approach would be to cluster the users based directly on their preferences and interests over entities in the HIN, rather than hoping that the matrix factorization will capture these preferences. To achieve this goal we employ a technique known as HyperEdge-Based Embedding (HEBE) [7], which learns distributed representations for entities in a heterogeneous information network based directly on the heterogeneous event data of which the network is composed. For example, an event in the IMDb network would represent a user watching a movie and would be represented by a hyperedge connecting the user, movie, and all other entities related to the movie such as actors, director(s), and the genre(s) of the movie. HEBE learns dense but low-dimensional representations for users based on the entities with which they have directly interacted, meaning that clustering users based on these representations will explicitly capture users' interests and preferences. After HEBE has learned user representations, we apply the *k-means* algorithm to find clusters of users using cosine distance. To make user recommendations, we use cosine similarity for $sim(C_k, u_i)$, meaning that each user may be described by several clusters if they are closely related in the learned vector space.

4.4 Model Learning

This chapter covers the learning algorithms for the global and personalized recommendation models discussed in Chapter 4. In a similar fashion, we introduce the learning method for the parameters of the global model from Equation 4.3 and then extend it to the personalized models expressed by Equation 4.4.

The models proposed in this work leverage the rich knowledge available in heterogeneous information networks by diffusing user feedback over a set of semantic relationships described by L different meta-paths. After preference diffusion, latent representations of users and items are estimated using NMF to capture the underlying preferences present in each diffused matrix. The learning problem is then to find weights with which to combine the user and item latent features in order to make the best recommendations. To do so, we use the implicit feedback from users as our training data, with the goal being for the model to accurately capture the actual behavior of the users. Recall that for implicit user feedback, a value of 1 means that a user was interested in an item, but a value of 0 is a mixture of items which the user is uninterested in and items which the user has not yet discovered. This means that traditional methods with classification or learning-to-rank based objectives, which treat values of 1 as positive and 0 as negative, are not well suited for the learning problem when using implicit user feedback and cannot generate high quality models.

Taking motivation from [22], we take a different learning approach and utilize a pairwise optimization strategy. Our objective function attempts to learn an ordering for items, in a pairwise manner, where values of 1 should be ranked higher than values of 0 for each user. The assumption underlying the proposed objective is that items with a value of 1 in the feedback data are more interesting to the corresponding users than all items with the value 0. This assumption is a weaker form of that used by the traditional methods, and is likely a closer model of reality which results in better performance.

4.5 Bayesian Ranking-Based Optimization

As is common among Bayesian methods, we aim to maximize the following posterior probability:

$$p(\theta|R) \propto p(R|\theta)p(\theta) \tag{4.5}$$

where $\theta = \{\theta_1, \theta_2, \dots, \theta_L\}$ are the global model parameters, and $p(R|\theta)$ represents the probability that all pairs of items for all users defined by R can be correctly ranked by the model. Therefore, a good model will have high $p(R|\theta)$ and be able to correctly rank items with the value 1 above items with the value 0 for each user.

Assuming that user preferences and the ordering of item pairs are independent allows us to expand the likelihood $p(R|\theta)$ as follows:

$$\begin{aligned} p(R|\theta) &= \prod_{u_i \in \mathcal{U}} p(R_i|\theta) \\ &= \prod_{u_i \in \mathcal{U}} \prod_{(e_a > e_b) \in R_i} p(e_a > e_b; u_i|\theta) \end{aligned} \tag{4.6}$$

where $(e_a > e_b) \in R_i$ are all pairs of items with correct ordering in R for user u_i and $p(e_a > e_b; u_i|\theta)$ represents the probability that user u_i prefers item e_a over e_b according to the model defined by θ. We define said probability as follows:

$$p(e_a > e_b; u_i|\theta) = \sigma(r(u_i, e_a) - r(u_i, e_b)) \tag{4.7}$$

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

We further assume that $p(\theta)$ is a Gaussian distribution with mean zero and variance-covariance matrix $\Sigma_\theta = \lambda I$. Using these definitions leads to the following objective function:

$$\begin{aligned} O &= -\ln p(\theta|R) = -\ln p(R|\theta)p(\theta) \\ &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \ln p(e_a > e_b; u_i|\theta) + \lambda||\theta||_2^2 \\ &= -\sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \ln \sigma(r(u_i, e_a) - r(u_i, e_b)) + \lambda||\theta||_2^2 \end{aligned} \tag{4.8}$$

where $\lambda||\theta||_2^2$ is a regularization term which depends on the data.

We estimate the parameters of the global model, θ, by minimizing O in Equation 4.8.

4.6 Optimization

Because Equation 4.8 is differentiable, there are many optimization strategies which we could apply to estimate the model parameters θ. We considered methods like stochastic gradient descent (SGD) [2] and L-BFGS-B [3]. To use any of these approaches, we first take the gradient of Equation 4.8 with respect to θ as follows:

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

Considering the scale of the data for real-world recommender systems, we chose to employ SGD [2] to estimate the parameters of our models in our experiments. We chose this method so as to avoid the $O(mn^2)$ time complexity needed to compute the full gradient, opting rather to compute the gradient using stochastic minibatches with a sampling rate of 10^{-5}. We discuss the sampling rate hyperparameter selection in Section 5.5.

Algorithm 4.1: Personalized Recommendation Model

Input: $R, G, M = \{\mathcal{P}_1, \ldots, \mathcal{P}_L\}, d, c, method$
Output: $\theta^{\{\cdot\}}$
for $q \leftarrow 1$ **to** L **do**
  foreach u_i and e_j **do**
    $\tilde{R}^{(q)}_{u_i,e_j} = s(u_i, e_j|\mathcal{P}^{(q)})$ (Eqn. 4.1)
  Calculate latent features $\hat{U}^{(q)}, \hat{V}^{(q)}$ from $\tilde{R}^{(q)}$ (Eqn. 4.2)

`/* Generate low-dimension user representations */`
if $method$ is HEBE **then**
  $U = \text{HEBE}(R, G, d)$
else
  $U, V = \text{NMF}(R, d)$
$C = \text{k-means}(U, c)$

`/* Learn recommendation models */`
foreach C_k in C **do**
  Optimize $\theta^{\{k\}}$ with implicit feedback of users in cluster C_k (Eqn. 4.8)

4.7 Learning Personalized Models

The proposed global model effectively leverages a heterogeneous information network to improve recommendation results, but it lacks the capacity to distinguish the underlying interests and preferences which drive user-item interaction for a particular user. Instead, the global model treats all users equally and finds the best one-size-fits-all solution to the problem. We observed that while users may have different interests, there exist groups of users which share common preferences and we can leverage this to learn more personalized models.

In Section 4.3 we covered two personalization approaches which can improve model performance. Beginning with NMF or HEBE, we produce user representations based on the implicit feedback or the entirety of the heterogeneous event data available, respectively. Then we apply the *k-means* clustering algorithm to cluster the users into c clusters. Finally, we learn a model for each of the clusters according to the learning method described above. The full learning algorithm for these models can be found in Algorithm 4.1.

Chapter 5

Experiments

In this chapter we present a variety of empirical studies of the proposed recommendation framework. We implemented the global model in Section 4.2, as well as both personalized models proposed in Section 4.3. For comparison, we also implemented several popular or state-of-the-art techniques for implicit feedback recommendation.

5.1 Data

We choose two real-world datasets from different domains for our empirical studies: movie recommendation using IMDb-MovieLens-100K (IM100K) data and local business recommendation using Yelp data. The network schemas for these two datasets can be found in Figure 5.3.

The IM100K dataset is a combination of the popular MovieLens-100k dataset with additional entities from IMDb related to the movies therein. Together, the MovieLens ratings with the entities from IMDb allow us to build a heterogeneous information network upon which we can apply our algorithms. While the MovieLens data contains explicit ratings, we treat it as implicit feedback data by assigning values of 1 for any movie with a user rated, and 0 for those which a user did not rate. In order to map movies and their associated entities from IMDb to MovieLens, we used their titles and release years, which could lead to a small percentage of errors. Consequently, the results we present below can be considered a lower-bound of the actual performance as a result of this source of noise.

The local business recommendation data comes from the Yelp challenge[1]. We did not need to augment this data as it contains both user reviews and local business information which can be fit into a heterogeneous information network. We build our implicit feedback by setting values of 1 for any user-business pair where the user wrote a review for the business, and 0 for all other pairs.

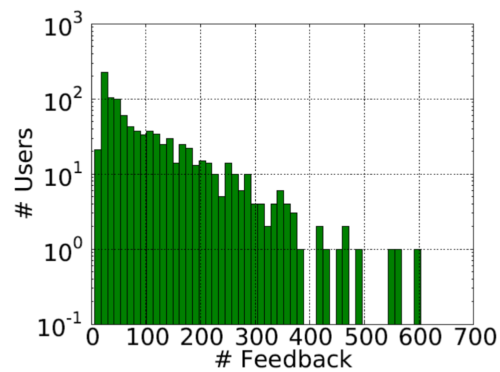
Each dataset has timestamp information along with the user interactions. We use these to split the feedback from each user into train and test sets in a fair manner, i.e., the model can only learn from a user's past interactions to predict their future interactions. This means that each user must have at least

[1] http://www.yelp.com/dataset_challenge/

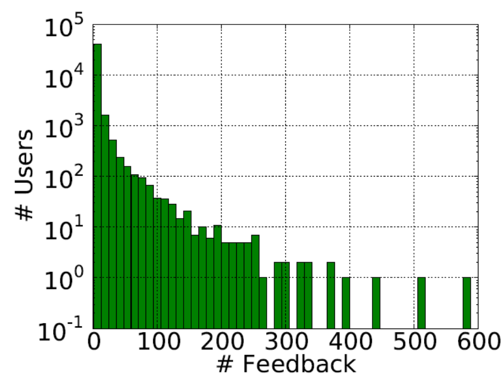
two interactions in order to be in both the train and test sets, so we filter out users with only only one interaction. We use a train/test split of 80%/20%.

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) Dataset Properties



(b) IM100K Feedback Distribution



(c) Yelp Feedback Distribution

Figure 5.1: Properties and feedback distributions for IM100K and Yelp Datasets

We summarize the properties of the two datasets in Figure 5.1a. Of note is the fact that the Yelp dataset is much sparser than the IM100K dataset, meaning that the performance of all methods degrades due to lack of signal. Additionally, the dataset's user feedback distributions maybe be found in Figure 5.1b and Figure 5.1c, demonstrating the power law behavior discussed in Section 4.3.

5.2 Baselines and Evaluation Metrics

As mentioned before, we implemented several popular or state-of-the-art recommendation techniques for baseline comparisons, their details are as follows:

- **Popularity:** Recommend the most popular items to all users.
- **Co-Click:** Estimate the 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, as discussed in Section 3.3.
- **Hybrid-SVM:** Use SVM-based ranking function [14] to learn a global recommendation model with user implicit feedback and meta-path based similarity measures [25].

Table 5.1: Example meta-paths (we set $n = 1$ and 2)

Network	Meta-Path
IM100K	$user$—$(movie$—$actor$—$movie)^n$ $user$—$(movie$—$director$—$movie)^n$ $user$—$(movie$—$genre$—$movie)^n$ $user$—$movie$—$keyword$—$movie$
Yelp	$user$—$(business$—$category$—$business)^n$ $user$—$(business$—$user$—$business)^n$ $user$—$business$—$checkin$—$business$ $user$—$business$—$location$—$business$

We refer to the proposed global model as *HeteRec-g* and the proposed personal models collectively as *HeteRec-p*. Unless otherwise specified, results we present results using the implicit feedback based personalization approach. We choose 10 different meta-paths in each information network, ranging from the most simple path of *user—item* to longer paths such as those introduced in Table 5.1.

For implicit feedback data, the standard evaluation metric of root mean squared error (RMSE) from recommendation models using explicit user feedback is not well suited. Consequently, we choose two well studied metric from information retrieval: Precision@k (P@k) and top-10 mean reciprocal rank (MRR) to evaluate the performance of each baseline as well as the proposed models. Precision@k is the average percentage of correct (i.e., appearing in the test data) recommendations in the top-k recommendations over all users. MRR is defined as follows:

$$MRR_K = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{e\in test(u_i)}\frac{1}{rank(u_i,r)}\right) \tag{5.1}$$

5.3 Performance Comparison

Table 5.2 presents the performance of each method across the two datasets.

Table 5.2: Algorithm Performance

Method	IM100K				Yelp			
	P@1	P@5	P@10	MRR	P@1	P@5	P@10	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	**0.2121**	**0.1932**	**0.1681**	**0.5530**	**0.0213**	**0.0171**	**0.0150**	**0.0513**

The simplest methods, popularity and co-click, perform moderately well in both datasets. It is no surprise that recommending the most popular items to all users can achieve moderate performance, due to the power law distribution of user feedback seen in Figure 5.1. The co-click method, which enjoys a very wide userbase, underperforms in the IM100K dataset but rivals NMF in the Yelp dataset where it scores an MRR of 0.0371 compared with NMF's 0.0382.

Moving to the more advanced collaborative-filtering-based matrix factorization method, NMF, we choose dimensionality $d = 20$ in IM100K and $d = 60$ in Yelp using cross validation on the training data. For fairness, we use the same method and settings in the preference diffusion step of the HeteRec-* methods. With further performance tuning and the inclusion of additional information as in [10], the performance of NMF may increase, but the same improvement would be enjoyed by our proposed models as well. We see that NMF is a strong contender, achieving MRR = 0.4938 in IM100K and 0.0382 in Yelp, outperforming the other baselines.

Hybrid-SVM, similar to our models, combines both implicit feedback and heterogeneous relations captured in a HIN. This hybrid recommendation approach uses the same amount of information as our proposed methods, but uses an SVM-based ranking framework and PathSim measures as features when learning a recommendation model. However, with the proposed diffusion and personalization strategies, Hybrid-SVM fails to fully leverage the richness of the data to improve performance. It achieves MRR = 0.4493 in IM100K and 0.0337 in Yelp, where it is even outperformed by co-click. When compared with our methods, which are based on the same amount of information, it is clear that the methods we have proposed are effectively leverage the data-right heterogeneous information networks to make entity recommendations.

Our global model, HeteRec-g, which uses the implicit user feedback augmented with a heterogeneous information network, is able to outperform all baselines. It improves upon the MRR of the strongest baseline, NMF, by 6.1% in IM100K and 10.4% in Yelp. This suggests that our assumptions that augmenting the implicit feedback with an information network can improve performance and help alleviate data sparsity by connecting users to movies via various meta-paths. Our method performs significantly better than Hybrid-SVM, which uses the same meta-paths and implicit feedback to define its own global model, proving that our preference diffusion strategy leads to latent features which correctly describe user preferences and interests. Interestingly, the MRR improvement of HeteRec-g over NMF is less for the more dense dataset IM100K than it is for the sparser dataset of Yelp. This aligns with the intuition that using information networks as we do can mitigate sparsity by bringing move items into close proximity to a user based on several longer paths in the network. During training, we use the same dimensionality as previous methods and a sample rate for SGD of 10^{-5}. We apply the same sample rate to all supervised approaches for these experiments.

Parameter tuning of the sample rate is discussed in Section 5.5.

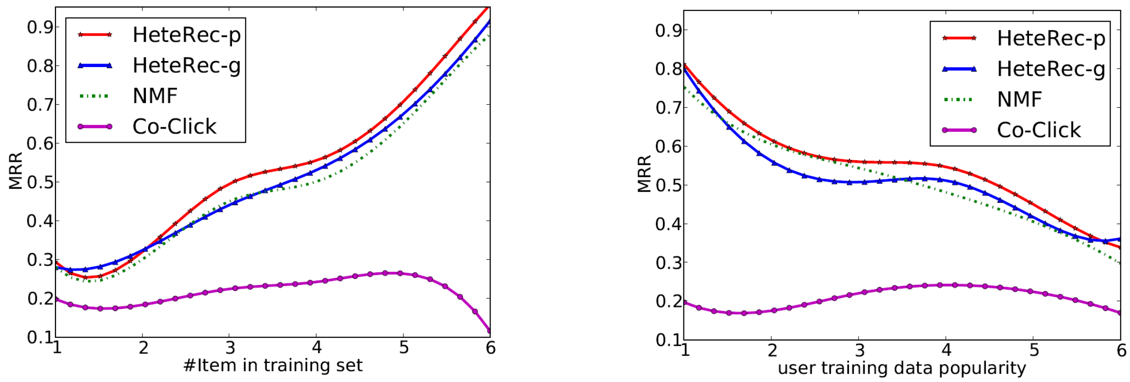
Out personalized recommendation approach, HeteRec-p, improves upon HeteRec-g further by treating learning user preference on a more granular level. For these experiments we used the implicit feedback based personalization strategy. We analyze the differences between the implicit feedback approach and the combined approach in Section 5.4. We use $c = 10$ in IM100K and $c = 100$ in Yelp. We discuss choosing the correct number of clusters, c, in Section 5.5. When comparing the personalized model with the global model, we see that HeteRec-p can outperform HeteRec-g in both datasets. The MRR improvement of HeteRec-p over HeteRec-g is 5.4% in IM100K and 21.5% in Yelp. This verifies our intuition that different users have different preferences and that the recommendation model should capture these preferences and recommend items to users based on their own interests, rather than treating all users identically. By learning recommendation models within user clusters which display similar interests in the feedback data, the model comes closer to approximating the human decision making process and as a result can offer each user higher quality recommendations.

Both HeteRec-g and HeteRec-p approached outperform all baseline models in both the IM100K and Yelp datasets. These experiments verify that using heterogeneous information networks to enrich the implicit user feedback, and learning more granular personalized models are both effective strategies for improving recommendation performance and satisfying users desires.

5.4 Performance Analysis

In order to understand when our models perform at their best and how they compare with others in these scenarios, we analyze the performance characteristics of co-click, NMF, HeteRec-g, and HeteRec-p (using implicit feedback only) when controlling the data for two variables: the average number of feedbacks given by a user and the average popularity of items on which a user has given feedback. We also discuss the performance difference between the two personalized models we propose.

When investigating the impact that feedback frequency has on algorithmic performance, we split users into groups based on the number of item interactions they had in the training data. We split the users into 6 groups, where users in group 1 had the least feedback (averaging 13) and those in group 6 had the most (averaging 224). We applied each of the 4 methods in each group, the results are displayed in Figure 5.2a. HeteRec-p outperforms the other methods in each group. As is expected, the performance of all methods is negatively impacted by the lack of information available for users in the lower groups. However, co-click is not nearly as influenced by this factor as the other methods, suggesting that the other CF based approaches



(a) Performance Difference by Feedback Count

(b) Performance Difference by Feedback Popularity

Figure 5.2: Performance Analysis

are much more sensitive to data sparsity and motivating our proposed use of heterogeneous information networks to help alleviate sparsity issues.

Next, we present the impact item popularity has on recommendation performance in Figure 5.2b. In this experiment we grouped users according to the average popularity of the items for which they gave feedback (i.e., the average number of users who rated the items they rated). As before, we split users into 6 groups, where the users in group 1 had an overall average item popularity of 71 and the users in group 6 had average item popularity of 281. This means that the users in group 1 tended to like less popular movies, and the users in group 6 tended to like much more popular movies. Again, HeteRec-p can be seen to be the winner overall, outperforming the other methods nearly across the board. We see that the CF based methods tend to perform best, in this experiment, when users prefer less popular items. This may seem unintuitive, but one explanation is that these users may be expressing actual interest which the models are able to capture, whereas users who prefer popular movies choose to watch anything that is popular rather than actually expressing any particular interests.

Finally, to understand the performance characteristics of the proposed personalization models we compare them both against one another and against the global model. Both personalized models outperform the global model overall. We can see that the personalization strategy which utilizes the extra information present in the heterogeneous information network to cluster users performs much better than the implicit feedback only approach according to the P@1 measure. This measure is perhaps the most important measure in many scenarios, where few recommendations are given and the quality of the topmost recommendation is of the utmost importance. While the HIN approach underperforms the implicit feedback approach in the other measures, the gap is not expansive meaning that the HIN approach could be employed to good effect on its own, or ideally merged with results from the implicit feedback only approach in order to get the best of both worlds. Such a combined approach would yield the highest performing model.

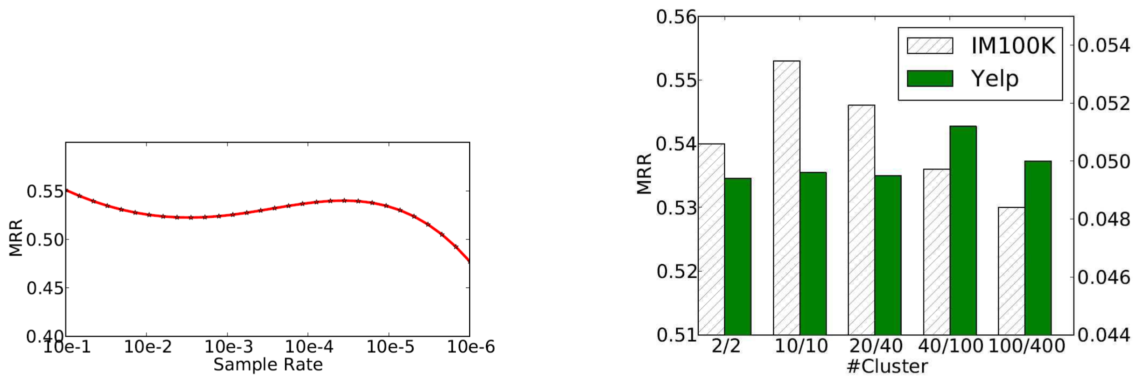
Table 5.3: Personalization Strategy Comparison

Method	IM100K			
	P@1	P@5	P@10	MRR
HeteRec-g	0.2094	0.1791	0.1614	0.5249
HeteRec-p (implicit only)	0.2121	**0.1932**	**0.1681**	**0.5530**
HeteRec-p (entire HIN)	**0.2312**	0.1801	0.1563	0.5377

5.5 Parameter Tuning

Our proposed methods include several hyperparameters that need to be tuned in order to achieve optimal performance. In this section we discuss their impact on the effectiveness of the proposed approaches and how to tune them.

In Equation 4.8, λ is a hyperparameter that adjusts the amount of L_2 regularization of the model parameters. We used cross-validation and grid search to set this parameter to 0.1.



(a) Sample Rate

(b) Cluster Count

Figure 5.3: Hyperparameter Tuning

The next hyperparameter of interest is the SGD sampling rate. This controls a tradeoff between exactness of the gradient updates and efficiency by only sampling a small portion of the entire training data to estimate the gradients at each step of training. If the sampling rate were set to 1 and the learning process scaled by $O(mn^2)$, the number of training instances in the Yelp dataset would be approximately 10^{12}, which is computationally unfeasible. We explore the relationship between the sampling rate and performance of HeteRec-g in IM100K in Figure 5.3a. The x-axis of Figure 5.3a is in log-scale, meaning that very small batches can be sampled without sacrificing too much overall performance. However, when the sampling rate falls below 10^{-5}, the performance does then to degrade as a result of a lack of training data for each step of the algorithm.

For the HeteRec-p models we have an additional hyperparameter c, the number of clusters for which to learn recommendation models. We investigate the performance of the model with respect to this hyperparameter in Figure 5.3b. Although the personalized model is not overly sensitive to this hyperparameter, it is clear that certain values lead to higher overall performance than others, with maximum values occurring at $c = 10$ for IM100K and $c = 100$ for Yelp. Clearly when the number of clusters is too small, the personalized models are unable to distinguish user interests and revert to something nearer to our proposed global model. By using an appropriately large number of clusters, different user behaviors are learned by the model and this results in better entity recommendations.

Chapter 6

Conclusion

In this thesis, we study the recommendation problem when faced with implicit user feedback. Our proposed models leverage data-rich heterogeneous information networks to both capture user interests and preferences and offer more personalized entity recommendations for users. User preferences are learned based on the semantic relationships described by meta-paths in the heterogeneous information networks, which when combined with the implicit user feedback data provide different views which each have the potential to explain the underlying motivation for user-item interactions. We introduce a global recommendation model which serves as the foundation for our proposed personalized models. We proposed two methods for personalization, leveraging both the implciit feedback data and the heterogeneous information network to give our model the capacity to learn why users are interested in the items they are. Using a Bayesian ranking method, we estimate the weights for both global and personalized models in an efficient manner. Comparisons between the proposed approaches and other widely used and state-of-the-art implicit feedback recommendation techniques show that our proposed approaches outperform other techniques. Several possible future works include model online model updating with user feedback, jointly learning a recommendation model along with user clustering, and an approximate learning process to further increase efficiency.

References

[1] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 9–, New York, NY, USA, 2004. ACM.

[2] L. Bottou. *Large-Scale Machine Learning with Stochastic Gradient Descent*, pages 177–186. Physica-Verlag HD, Heidelberg, 2010.

[3] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, Sept. 1995.

[4] S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 571–580, New York, NY, USA, 2007. ACM.

[5] C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):45–55, Jan. 2010.

[6] Q. Gu, J. Zhou, and C. H. Q. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 199–210, 2010.

[7] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han. Large-scale embedding learning in heterogeneous event data. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 907–912. IEEE, 2016.

[8] 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 *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, pages 53–60, 2009.

[9] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 259–266, 2003.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272. IEEE Computer Society, 2008.

[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In X. Amatriain, M. Torrens, P. Resnick, and M. Zanker, editors, *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pages 135–142. ACM, 2010.

[12] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 538–543. ACM, 2002.

[13] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In C. Apté, J. Ghosh, and P. Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1298–1306. ACM, 2011.

[14] T. Joachims. Training linear svms in linear time. In T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 217–226. ACM, 2006.

[15] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Y. Li, B. Liu, and S. Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 426–434. ACM, 2008.

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

[17] T. Kuo, R. Yan, Y. Huang, P. Kung, and S. Lin. Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 775–783. ACM, 2013.

[18] N. Lao and W. W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.

[19] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 203–210. ACM, 2009.

[20] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K. Choi, and A. Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 931–940. ACM, 2008.

[21] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In R. Dechter and R. S. Sutton, editors, *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada.*, pages 187–192. AAAI Press / The MIT Press, 2002.

[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In J. A. Bilmes and A. Y. Ng, editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461. AUAI Press, 2009.

[23] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In L. D. Raedt and S. Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 713–719. ACM, 2005.

[24] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In V. Y. Shen, N. Saito, M. R. Lyu, and M. E. Zurko, editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 285–295. ACM, 2001.

[25] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[26] 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 Q. Yang, D. Agarwal, and J. Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1348–1356. ACM, 2012.

[27] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 797–806. ACM, 2009.

[28] 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.

[29] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, pages 1119–1130. SIAM / Omnipress, 2012.

[30] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA*, 27, 2013.

[31] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 347–350. ACM, 2013.

[32] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 245–252. ACM, 2011.